ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI
Radmile Matejčić 2, 51000 Rijeka, www.inf.uniri.hr

CURRICULUM OF THE UNDERGRADUATE UNIVERSITY STUDY PROGRAMME IN

INFORMATICS

Rijeka, 2019

1. INTRODUCTION

1.1. REASONS FOR INITIATING THE STUDY PROGRAMME

The study programme in Informatics has been offered at the Department of Informatics, University of Rijeka since the department was founded back in 2008. However, the history of studying informatics at the University of Rijeka dates back even further. Starting from 1975, it was first carried out as a two-year, and later as a four-year study programme in Informatics at the then Faculty of Industrial Education. In the year 1984/1985, the study programme in mathematics and informatics was launched at the then Faculty of Education, where the Institute for Informatics was founded in 1987. Its aim was to bring together all the material from the field of informatics and to modernise the teaching content for this study programme by applying information technology. The Institute for Informatics was renamed Department of Informatics in 1994. First it formed part of the Faculty of Education, and since 1998 of the newly founded Faculty of Humanities and Social Sciences in Rijeka. In the academic year 1999/2000, a double-major study programme in Informatics was launched. It was designed as an open programme that could be combined with other study programmes at the Faculty of Humanities and Social Sciences such as philosophy, history, pedagogy, English language, German language etc. In 2004, a double-major in physics and informatics was also launched. A permit for the independent single-major undergraduate study programme in Informatics, aligned with the principles of the Bologna process, was issued in the academic year 2005/2006.

After the Department of Informatics, University of Rijeka was founded in 2008, all permits for carrying out the said study programmes held by the former department at the Faculty of Humanities and Social Sciences were transferred to the Department as its legal successor. In its proposed form, with minor modifications introduced in the academic year 2012/2013, the study programme was carried out until the academic year 2018/2019, which saw the intensification of work on its modifications started in 2017 as part of the Stand4INFO project.

The Department of Informatics was a partner in the project "Stand4INFO – Development of Higher Education Standards of Professional Interest, Qualification Standards and Study Programmes based on the Croatian Qualifications Framework in the Field of Informatics" (http://stand4info.foi.hr/), which was part of the ESF project "Improvement of Higher Education Quality with the Application of the Croatian Qualifications Framework", coordinated by the Faculty of Organisation and Informatics (FOI) of the University of Zagreb. In included joint work on the standards of professional interest and qualifications, as well as on the revision of study programmes, including this one.

Amendments to the existing study programme were necessary in order to align it with the requirements of the labour market, which follows the dynamics and rapid development of ICT, and to increase the quality and modularity of the study programme. They were introduced with the aim of aligning the study programme with the future qualification standard for the university bachelor of informatics, which is in the process of being entered in the CROQF register. The percentage of modified ECTS credits is rather high, but the objectives of the majority of courses within the existing study programme have not changed significantly, so we can talk about the modernisation of the existing programme.

The proposed study programme consists of the compulsory part, equal for all students, and elective modules that students may choose according to their own preferences and interests. Learning outcomes for compulsory courses (first two years of the study programme and courses Bachelor's Thesis and Undergraduate Internship in the last semester of the 3rd year) have been aligned with units of learning outcomes (ULOs) for the proposed qualification standard for the "bachelor of informatics" and assigned appropriate assessments. This qualification standard corresponds to the occupation of software engineer from the CROQF occupational standard subregister (https://hko.srce.hr/registar/standard-zanimanja/detalji/11).

The modules are introduced in the 3rd year of studies as sets of related courses. They are intended to cover a wide range of different jobs expected on the present-day labour market from "information scientists", i.e. software engineers who completed undergraduate study programmes in informatics. They also allow students to specialize in those fields that correspond to their preferences. When designing the modules, special attention was paid to the requirements of the labour market as well as to the practices of similar study programmes in this field. As a result, 4 modules have been proposed:

1. Software Development (SD)
2. Communication Systems (CS)
3. Multimedia Systems (MMS)
4. Information Systems (IS)

In the 3rd year of their studies, students choose one of the modules, thus specializing in the chosen field of informatics. In addition, students can choose compulsory courses from other modules as their elective courses or they can choose shared elective courses, which also increased in number compared to the previous version of the study programme. Elective courses include various courses in mathematics that were compulsory in the previous version of the programme. Since their outcomes do not fall under compulsory learning outcomes according to the future CROQF standard for the university bachelor of informatics, students can now choose them as elective courses. Another novelty is the possibility of enrolling into one of the shared elective courses offered by the University of Rijeka and its constituents, with a minimum of 3 ECTS credits.

1.2. EVALUATION OF RATIONALITY OF THE STUDY PROGRAMME AND ALIGNMENT WITH THE INSTITUTIONAL STRATEGY FOR THE DEVELOPMENT OF STUDY PROGRAMMES

Given the interdisciplinary nature of informatics, according to CROQF some occupations from the field of informatics belong to sector XVIII Information and Communication, and some to sector VII Electrical Engineering and Computing (subsector Computing). The status and possibilities for the development of Information and Communication Sector are elaborated in more detail in the Industrial Strategy of the Republic of Croatia 2014–2020, according to which the two prevailing activities in the ICT industry, namely computer programming, consultancy and related services and telecommunications generated almost three fourths of the added value in the industry and employed almost two thirds of the total number of employees in the industry. The Industrial Strategy highlights the lack of educational force as the biggest problem of the ICT industry already at this stage, and especially in near future, as demand for IT professionals will continue and ICT will represent the main generator of economic and social development. For this reason, it is important to have modern study programmes that will form future IT professionals, such as the proposed undergraduate study programme in informatics, offering competencies corresponding to the occupation of software engineer.

The new version of the study programme, and especially the introduction of modules and Undergraduate Internship as a new course, are aligned with the objectives and tasks of the Strategy of the University of Rijeka 2014–2020, and especially with:

IIe – 2 Listing practical competencies guaranteed on completion of degree programmes and harmonising degree programmes in the process or (re)accreditation,

and also with:

IIa - 4.2a Increasing the number of students in technology, biology, medicine, biotechnology and natural sciences, in information and communication, and in related interdisciplinary fields

IIb - 1.1 Increasing the student satisfaction index

IId – 1 Increasing the number of students in the incoming and outgoing mobility system.

When we talk about increasing the quality of studies, study programme satisfaction surveys showed that students at the Department of Informatics pointed out the following aspects of studies as least satisfactory: “Possibility of gaining practical competencies”, “Number of elective courses” and “Training for jobs in the profession”. By modernising the study programme, introducing the course Undergraduate Internship and offering the choice of not only a larger number of elective courses, but also modules, the quality of study programme will improve, as well as the satisfaction of enrolled students, which will also enable larger enrolment quotas.

The planned duration of each course is one semester, which enables dynamic exchange of content. It also allows students to go to other domestic or foreign universities at any stage of their study programme, and enables visiting students to come through mobility and student exchange schemes, such as Erasmus+.

1.3. COMPARISON OF THE STUDY PROGRAMME WITH SIMILAR PROGRAMMES AT ACCREDITED HIGHER EDUCATION INSTITUTIONS IN THE REPUBLIC OF CROATIA AND EU

Courses in informatics, under same or related names, but with similar content are found in most of European and American study programmes. Basic topics they usually tackle include: programming, operating systems, Internet, computer networks, multimedia, databases, computer architecture etc.

During the undergraduate study programme, students acquire fundamental scientific knowledge forming the basis for IT knowledge, such as: fundamentals of digital technology and computer architecture, operating systems, programming using algorithms and data structures, computer networks and Internet, databases, information systems, object-oriented programming and programming for the Internet, data and process modelling, fundamentals of formal languages and creation of multimedia applications. During the undergraduate study programme, students also acquire mathematical knowledge necessary for the study programme in informatics. This is achieved through four courses in mathematics, with the possibility of choosing additional elective courses.

The modified study programme, same as the original version of the programme, corresponds to the curriculum of the undergraduate study programme Information Systems at the Faculty of Organisation and Informatics of the University of Zagreb – FOI (https://www.foi.unizg.hr/studiji/pds/ips#informacijski-sustavi), that is, it represents a modernised version of this study programme.

Although the study programme, according to the classification in the Republic of Croatia, belongs to social sciences, certain elements related to the modules or specific courses are also aligned with technical study programmes: undergraduate study programme “Computer and Information Science” at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia (https://fri.uni-lj.si/en/study-programme/computer-and-information-science), undergraduate study programme “Computer Science” at the Graz University of Technology, Austria (https://www.tugraz.at/studium/studienangebot/bachelorstudien/informatik/) and undergraduate study programme “Computing” at the Faculty of Electrical Engineering and Computing, University of Zagreb – FER (https://www.fer.unizg.hr/studiji/fer3/racunarstvo).

2. GENERAL INFORMATION

2.1. NAME OF THE STUDY PROGRAMME

Undergraduate study programme in informatics

2.2. PROVIDER AND INSTITUTION IMPLEMENTING THE STUDY PROGRAMME

University of Rijeka, Department of Informatics,
Radmile Matejčić 2, 51000 Rijeka

2.3. TYPE OF THE STUDY PROGRAMME

Undergraduate university programme

2.4. DURATION OF THE STUDY PROGRAMME

Anticipated duration of the Undergraduate study programme in informatics is three academic years, i.e. six semesters.

Upon completion, the student acquires a minimum of 180 ECTS credits.

2.5. ENROLMENT REQUIREMENTS

Candidates who have completed four-year secondary education and passed the nationwide leaving exam according to applicable requirements, and pursuant to the law, are entitled to respond to the call for applications for the Undergraduate university study programme in informatics. The choice of candidates for enrolment into the undergraduate university study programme is made based on the grades achieved in the secondary school and results of nationwide leaving exam.

The right to apply is granted to the citizens of the Republic of Croatia, foreign citizens and stateless persons.

2.6. COMPETENCIES

Competencies

Upon completion of the Undergraduate study programme in informatics, students gain competencies from the sets of competencies corresponding to the occupation of software engineer in the CROQF occupational standard subregister (https://hko.srce.hr/registar/standard-zanimanja/detalji/11)

Sets of competencies:

1. Cooperation and communication during software development
2. Modelling, restructuring and enhancing business processes

3. Database design
4. Software development
5. Designing technical and communication infrastructure and information systems
6. Writing software documentation
7. Information system integration
8. Testing information systems and their components
9. Software maintenance
10. Personal and professional development in the IT field
11. Enhancing the development process and workload organisation in the IT field

Key tasks

Following the completion of the study programme, students acquire knowledge and skills that will enable them to perform the following key tasks (according to the list of key tasks corresponding to the occupation of software engineer in the CROQF occupational standard subregister):

1. Developing and testing software components
2. Integrating and testing software solutions
3. Adapting, implementing and maintaining software
4. Supporting development activities in informatics
5. Documenting software and individual software components

Possibility to continue education (at a higher level)

A student who has completed the Undergraduate study programme in informatics can enrol into the graduate study programme in informatics at all universities in Croatia and worldwide. They may be required to take supplemental exams.

2.7. LOCATION AND EQUIPMENT

Since October 2012, the Department of Informatics of the University of Rijeka operates in a building situated within the University Campus on Trsat. The facilities used by the Department account for 14.86% of net surface area of the entire object (1,411.73 m2) intended for university departments. In addition to our department, the building is also home to the Department of Mathematics, Department of Physics and Department of Biotechnology, Campus Branch of the University Library Rijeka, CARNet, Center for Learning and Teaching Support, University IT Centre, Centre for Advanced Computing and Modelling etc.

Out of the total surface area of the Department, classrooms, laboratories and demonstration rooms take up 837.18 m2, professors' offices account for 574.55 m2 and library takes up 313 m2. The Department also has at its disposal two large lecture halls that can accommodate 100 and 150 students each, as well as two smaller ones that can accommodate 40 to 50 students. We currently have 3 computer classrooms with 83 modern computers, and from the beginning of the academic year 2019/2020 we will have another computer classroom with 36 new computers, which means almost 120 computers in total for our classes. In addition, the Department has 2 laboratories with state-of-the-art computer equipment and a room used by doctoral students for their work and research. Undergraduate and graduate students have at their disposal an additional computer classroom for individual work and training, which holds 15 computers. The library within the building is also equipped with computers, so this is another place where students can do their homework, write their papers, do their research etc.

All classrooms have fixed LCD projectors and a computer to which they are connected. The equipment is used during classes, but students can also use it outside class time with their AAI identity. Computer classrooms also have power sockets and network ports to which students can connect their laptops. Wireless network is available in classrooms, but also in other parts of the building.

Computer equipment used by the Department employees in their work (computers in their offices) has been procured in the past years so as to ensure that none of them is older than 5 years. Funds for the equipment are ensured from the revenues of the Department. Professors also get the necessary equipment for their scientific research work, such as tablets, laptops etc. from the funds obtained for scientific projects.

2.8. USE OF E-LEARNING SYSTEM

The study programme will use e-learning technologies and methods for blended or fully online teaching.
All courses will be included as e-courses in one of the learning management systems. For example, in Merlin system by SRCE, which is based on the Moodle open-source system .

2.9. PROFESSIONAL OR ACADEMIC TITLE OR DEGREE AWARDED UPON COMPLETION OF THE STUDY PROGRAMME

Bachelor of informatics.

3. DESCRIPTION OF THE STUDY PROGRAMME

3.1. LIST OF COMPULSORY AND ELECTIVE COURSES AND/OR MODULES WITH THE NUMBER OF CLASS HOURS REQUIRED FOR THEIR IMPLEMENTATION AND THE NUMBER OF ECTS CREDITS

LIST OF MODULES/COURSES							
Semester: 1							
MODULE	COURSE	COURSE INSTRUCTOR	L	E	S	ECTS	STATUS[1]
All	Mathematics 1	Asst. Prof. Milena Sošić, PhD	2	2	0	6	C
All	Programming 1	Full Prof. Maja Matetić, PhD	2	2	0	6	C
All	Fundamentals of Informatics	Full Prof. Sanda Martinčić-Ipšić, PhD	2	2	0	5	C
All	Multimedia Systems	Full Prof. Nataša Hoić-Božić, PhD	2	2	0	5	C
All	Fundamentals of Economics for IT students	Asst. Prof. Borna Debelić, PhD	2	2	0	5	C
All	English Language for IT Profession	Irena Grubica, MSc / Lucia Načinović Prskalo, PhD	2	2	0	4	C
All	Physical Education 1						C
LIST OF MODULES/COURSES							
Semester: 2							
MODULE	COURSE	COURSE INSTRUCTOR	L	E	S	ECTS	STATUS[2]
All	Mathematics 2	Asst. Prof. Ana Jurasić, PhD	2	2	0	5	C
All	Programming 2	Full Prof. Maja Matetić, PhD	2	2	0	5	C
All	Operating Systems	Assoc. Prof. Božidar Kovačić, PhD	2	2	0	5	C
All	Data Modelling	Full Prof. Mile Pavlić, PhD	2	2	0	5	C
All	Computer Architecture and Organization	Full Prof. Ivo Ipšić, PhD	2	2	0	5	C
All	Fundamentals of Probability and Statistics	Asst. Prof. Davor Dragičević, PhD	2	2	0	4	C
All	Physical Education 2						C
LIST OF MODULES/COURSES							
Semester: 3							
MODULE	COURSE	COURSE INSTRUCTOR	L	E	S	ECTS	STATUS[3]
All	Mathematics 3	Asst. Prof. Marija Maksimović, PhD	2	2	0	5	C
All	Business Process Analysis	Full Prof. Mile Pavlić, PhD	2	2	0	5	C

[1] IMPORTANT: Insert **C** if the course is compulsory or **E** if the course is elective.

[2] IMPORTANT: Insert **C** if the course is compulsory or **E** if the course is elective.

[3] IMPORTANT: Insert **C** if the course is compulsory or **E** if the course is elective.

All	Object-Oriented Programming	Assoc. Prof. Marina Ivašić-Kos, PhD	2	2	0	5	C
All	Computer Networks	Full Prof. Mario Radovan, PhD / Vedran Miletić, PhD	2	2	0	5	C
All	Databases	Full Prof. Patrizia Poščić, PhD	2	2	0	5	C
All	Introduction to Software Engineering	Assoc. Prof. Sanja Čandrlić, PhD	2	2	0	5	C
LIST OF MODULES/COURSES							
Semester: 4							
MODULE	COURSE	COURSE INSTRUCTOR	L	E	S	ECTS	STATUS[4]
All	Operations Research	Asst. Prof. Martina Holenko Dlab, PhD	2	2	0	5	C
All	Introduction to Web Programming	Full Prof. Mario Radovan, PhD / Igor Jugo, PhD	2	2	0	5	C
All	Algorithms and Data Structures	Asst. Prof. Marija Brkić Bakarić, PhD	2	2	0	5	C
All	Information Technology Project Management	Full Prof. Sanda Martinčić Ipšić, PhD	2	2	0	5	C
All	Information Systems Development	Full Prof. Mile Pavlić, PhD	2	2	0	5	C
All	Security of Information and Communication Systems	Assoc. Prof. Božidar Kovačić, PhD	2	2	0	5	C
LIST OF MODULES/COURSES							
Semester: 5							
MODULE	COURSE	COURSE INSTRUCTOR	L	E	S	ECTS	STATUS[5]
SD	Programming Paradigms and Languages	Assoc. Prof. Marina Ivašić-Kos, PhD	2	2	0	5	C
SD	Code Optimization	Asst. Prof. Miran Pobar, PhD / Vedran Miletić, PhD	2	2	0	5	C
SD	Web Programming	Assoc. Prof. Božidar Kovačić, PhD / Igor Jugo, PhD	2	2	0	5	C
CS	Communication Networks	Full Prof. Mario Radovan, PhD / Vedran Miletić, PhD	2	2	0	5	C
CS	Network and Mobile Operating Systems	Assoc. Prof. Božidar Kovačić, PhD	2	2	0	5	C
CS	Social Network Analysis	Assoc. Prof. Ana Meštrović, PhD	2	2	0	5	C
MMS	Multimedia Technologies	Asst. Prof. Miran Pobar, PhD	2	2	0	5	C
MMS	Computer Graphics	Asst. Prof. Martina Holenko Dlab, PhD	2	2	0	5	C
MMS	Computer Animation	Full Prof. Nataša Hoić-Božić,	2	2	0	5	C

[4] IMPORTANT: Insert **C** if the course is compulsory or **E** if the course is elective.

[5] IMPORTANT: Insert **C** if the course is compulsory or **E** if the course is elective.

		PhD / Vanja Slavuj, PhD					
IS	Database Administration and Security	Full Prof. Patrizia Poščić, PhD	2	2	0	5	C
IS	User Interface and Interaction Design	Assoc. Prof. Sanja Čandrlić, PhD	2	2	0	5	C
IS	Information systems for specific purposes	Assoc. Prof. Sanja Čandrlić, PhD / Martina Ašenbrener Katić, PhD	2	2	0	5	C
SD/CS/ MMS/IS	Elective courses (from other modules or common)		2	2	0	15	E
Common elective courses							
	System Dynamics	Asst. Prof. Marija Brkić Bakarić, PhD	2	2	0	5	E
	Programming for Data Science	Full Prof. Maja Matetić, PhD	2	2	0	5	E
	Introduction to Theoretical Computing	Full Prof. Sanda Martinčić-Ipšić, PhD	2	2	0	5	E
	Combinatorics	Full Prof. Sanja Rukavina, PhD	2	2	0	5	E
	Numerical Mathematics	Asst. Prof. Bojan Crnković, PhD	2	2	0	5	E
	Mathematical Logic	Asst. Prof. Tajana Ban Kirigin, PhD	2	2	0	5	E
LIST OF MODULES/COURSES							
Semester: 6							
MODULE	COURSE	COURSE INSTRUCTOR	L	E	S	ECTS	STATUS[6]
All	Bachelor's Thesis	Assoc. Prof. Ana Meštrović, PhD				8	C
All	Undergraduate Internship	Asst. Prof. Martina Holenko Dlab, PhD				4	C
All	Elective course (common elective course offered by the University of Rijeka/course from another module/common elective course)	University of Rijeka				3	E
SD	Desktop and Mobile Application Development	Asst. Prof. Marija Brkić Bakarić, PhD	2	2	0	5	C
SD	Declarative Programming Languages	Assoc. Prof. Ana Meštrović, PhD	2	2	0	5	C
CS	Introduction to Embedded Systems and the Internet of Things	Asst. Prof. Miran Pobar, PhD	2	2	0	5	C
CS	Computer System	Full Prof. Ivo Ipšić, PhD /	2	2	0	5	C

[6] IMPORTANT: Insert **C** if the course is compulsory or **E** if the course is elective.

	Administration	Vedran Miletić, PhD					
MMS	Fundamentals of Game Development	Assoc. Prof. Marina Ivašić-Kos, PhD	2	2	0	5	C
MMS	Multimedia Design	Full Prof. Nataša Hoić-Božić, PhD	2	2	0	5	C
IS	Introduction to Data Analysis and Visualization	Full Prof. Sanda Martinčić-Ipšić, PhD / Lucia Načinović Prskalo, PhD	2	2	0	5	C
IS	Next Generation Databases	Full Prof. Patrizia Poščić, PhD / Danijela Jakšić, PhD	2	2	0	5	C
SD/CS/ MMS/IS	Elective course (from other modules or common)		2	2	0	5	E
Common elective courses							
	Programming for Complex Problem Solving	Assoc. Prof. Ana Meštrović, PhD	2	2	0	5	E
	Discrete Mathematics	Full Prof. Dean Crnković, PhD	2	2	0	5	E
	Set Theory	Asst. Prof. Tajana Ban Kirigin, PhD	2	2	0	5	E

3.2. COURSE DESCRIPTION

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Milena Sošić, PhD	
Name of the course	Mathematics 1	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	6
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master basic concepts and results of basic mathematics (sets, relations, functions) and linear algebra (matrices, determinants, system of linear equations), which are absolutely vital for the application of mathematical knowledge in information sciences. Furthermore, the objective of the course is to encourage and train students to use logical reasoning and apply their mathematical knowledge in science and economy.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

I1. Describe the basics of mathematical modelling and formulation of formal mathematical theory, distinguish between concepts of definition and theorem as well as between usual types of mathematical proof.

I2. Analyse and formally write mathematical statements using appropriate conventional propositional logic formulas or quantification logic formulas.

I3. Analyse relations and operations between sets.

I4. Define and analyse properties of binary relations, including equivalence and order relations, and apply them to solve problems from the field of information sciences.

I5. Present basic concepts and theorems of matrix calculus.

I6. Apply matrix calculus to solve standard mathematical problems and problems from the field of information sciences.

I7. Demonstrate basic theorems on the solvability of systems of linear equations.

I8. Model an actual problem using the system of linear equations and inequalities and choose the right method for solving the system of linear equations and inequalities.

4. Course content

- Fundamentals of mathematical modelling in science. Structure of presenting a mathematical theory and standard types of mathematical proof.
- Mathematical induction method.
- Syntax and semantics of propositional logic. Basics of quantification logic.

- Sets, set operations.
- Classification of binary relations. Equivalence relations. Order relations.
- Functions. Function composition. Inverse functions.
- Equipotent sets. Finite and infinite sets. Cardinality.
- Concepts of matrix addition and multiplication, matrix rank and inverse matrix.
- Determinant and properties of determinants.
- System of linear equations. Existence of solution. General solution of the linear system of equations. Gaussian elimination.
- System of linear inequalities.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online assignments and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes and participate in course activities within the distance learning system.
- Participate in assessments (tests and preliminary exams) and obtain enough credits for taking the exam. Minimum number of credits (per activity) necessary to take the final exam is specified in the detailed course syllabus.
- In the final exam, present the acquired material in a well-argued manner according to the content of the course. Passing score for the exam is 50%.

A detailed scoring system for the course, including individual passing scores, will be specified in the course syllabus.

8. Monitoring[7] of student work

Class attendance	1	Class participation	1	Seminar paper		Experimental work	
Written exam		Oral exam	2	Essay		Research	
Project		Continuous assessment	2	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through short tests, preliminary exams and an oral exam.

[7] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- In the written or online assessment (tests), students demonstrate their understanding of theoretical concepts from the course material (I3, I4, I5, I6) through multiple choice questions, fill in the blank questions and essay questions. For example, provide an example of binary relation, which is an equivalence relation, with the explanation of necessary properties, and determine the corresponding equivalence class and partition.
- In the written assessment (preliminary exams), students demonstrate that they have mastered the course material through mathematical modelling of a problem, application of knowledge to specific assignments, analysis of properties and choice of methods for solving the problem (I2, I3, I4, I5, I6, I8). For example, write a given sentence using a propositional logic formula and establish the corresponding normal forms and test their validity, satisfiability and refutability.
- In the written and oral assessment (final exam), students demonstrate that they have mastered the theoretical concepts of the course material through mathematical modelling of a problem, formulation of mathematical statements, analysis of properties and discussion of examples and by proving them using an appropriate method (I1, I2, I3, I4, I5, I6, I7). For example, recognize order relations among offered examples of binary relations, and explain the necessary properties.

10. Mandatory literature (at the time of submission of study programme proposal)

1. M. Radić, Algebra I dio, Školska knjiga, Zagreb, 1989.
2. K. Horvatić, Linearna algebra, Tehnička knjiga, Zagreb, 2004.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. M. Sošić, M. Marinović, Repetitorij s riješenim zadacima iz matematike, Filozofski fakultet, Rijeka, 2004.
2. B. Divjak, T. Hunjak, Matematika za informatičare, TIVA, Fakultet organizacije i informatike, Varaždin, 2004.
3. B. Divjak, T. Hunjak, Zbirka zadataka iz matematike, TIVA, Fakultet organizacije i informatike, Varaždin, 2002.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Maja Matetić, PhD	
Name of the course	Programming 1	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	6
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The course provides basic understanding of programming approaches, concepts and procedures and gives an introduction to modular software structure. The course includes topics related to algorithm development and implementation processes, use of language constructs in a simple programming code and methods of code debugging. The course introduces students to some commonly used algorithms in a given imperative programming language.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Design basic components of an algorithm, concretize them up to the level of procedural algorithm, and write them down using pseudocode or activity diagram.

I2. Choose appropriate types of data for a given problem and efficiently store the input data and implement the chosen organisation of data using a given imperative programming language.

I3. Adapt and optimise the algorithm solution given as a pseudocode or activity diagram according to the specificities of programming constructs (selection, iteration...) of a given imperative programming language.

I4. Choose parts of algorithm that can be isolated into separate units and executed as subroutines and implement them using a given imperative programming language.

I5. Build own subroutine libraries with commonly used subroutines and include them into programs if necessary.

I6. Choose appropriate log structure for storing data into files and implement it using a given programming language.

I7. Interpret error and warning messages returned by the compiler and fix the given program accordingly.

I8. Choose test samples of input data for testing a given program for usual and marginally permitted instances of a given problem.

4. Course content

Basics of C++ (variables and assignment, input and output, types of data and expressions). Execution flow control (if-else statement, simpler use of loops: while, do-while, program style). Multiway branch in a programme (logical expressions, nested if statement, switch statement). More complex use of loops: while, do-while, for. Loop design (loop exit, nested loops, loop troubleshooting). Array. Structures. String. Functions (value-returning functions, type casting, functions defined by the programmer, procedural abstraction, local variables). Void functions. Reference value forwarding. Function libraries.

Functional testing. Troubleshooting techniques. Function overriding. Files. Directories.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system, prepare for classes (do homework) and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical quizzes and preliminary exams) and successfully pass them.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[8] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Preliminary exams	2	Quizzes	1.5		

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Practical assessment on a computer (participation in solving assignments in class, answering questions) (I1). For example, answering questions related to the design of basic algorithm components.
- Written or online assessment of theoretical knowledge (quizzes) (I2, I3). For example, assessment of

[8] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

theoretical knowledge through multiple choice questions, fill in the blank and/or essay questions regarding types of data and programming language constructs (sequential flow of program execution, selection, iteration).

- Practical assessment on a computer (homework), where students solve simple problems by designing an algorithm for problem solving and its implementation in a chosen programming language (I3, I4).
- Practical assessment (homework and preliminary exam), where students need to build their own subroutine libraries with commonly used subroutines and include them into a program and implement file storage (I5, I6).
- Practical assessment (homework and preliminary exam), in which students need to know how to interpret error messages and fix the programme accordingly (I7, I8). For example, to test a program, students need to prepare test samples.
- Practical assessment (final exam), where students apply the acquired programming skills and knowledge to solve assignments according to the set instructions and evaluation criteria (I2, I3, I4, I5, I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Julijan Šribar, Boris Motik: Demistificirani C++, Dobro upoznajte protivnika da biste njime ovladali, Element, Zagreb, 2001.
2. Maja Matetić: Skripta uz predmet Programiranje 1 (digitalna skripta), Odjel za informatiku, Sveučilište u Rijeci, Rijeka 2012.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Nina Lipljin: Programiranje/1, TIVA Tiskara Varaždin, 2004.

2. Vulin, R.: Zbirka riješenih zadataka iz C-a, Školska knjiga, Zgb, 2003.

3. Walter Savitch: Problem Solving in C++, Pearson Publishing, 2006.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Sanda Martinčić-Ipšić, PhD	
Name of the course	Fundamentals of Informatics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge from the field of information and communications technology, basic principles of computer operation and basics of programming paradigms.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe basic components of a computer system: hardware and numerical formats.

I2. Apply information and communications technologies when solving semi-complex problems related to data collection, processing and presentation using office management applications.

I3. Choose and apply basic Internet services (e-mail, information search, use and set-up of web browser etc.).

I4. Recognise and express development trends in the information and communications technology.

I5. Name and group properties of different types of licences in the field of software development and application.

I6. Recognize and group basic properties of software and programming languages.

I7. Use basic program structures, control elements and data structures in a programming language (Python).

I8. Apply regular expressions in programs for simple business data processing (Python).

4. Course content

- Introduction to informatics, overview of concepts and definitions.
- Historical overview of the development of hardware, computer paradigms, operating systems and programming languages.
- Introduction to computer structure. Command execution on a computer. Numerical formats and representations.
- Introduction to theoretical ~~basis of~~ computing. Finite-state automata and regular expressions.
- Properties and types of programming languages.
- Introduction to Python, command syntax. Data structures. Functions, regular expressions. Packages. Visualization.

- Software licensing.
- Development trends in the information and communications technology.
- Writing of academic papers, creation of (business) presentations.
- Use of applications and basic Internet services for office operations.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	☒ fieldwork	□ other ___________________

6. Comments	Classes are held by combining classroom work and computer laboratory work with individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule of lectures and exercises will be specified in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system. Fieldwork is organised according to the possibilities (e.g. visit to the Peek&Poke museum).

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow and take part in course activities.
- Participate in continuous assessment (preliminary exams).
- Do homework and complete individual assignments during exercises.
- Independently write a seminar paper according to the instructions on academic writing and prepare a presentation of the paper.
- Complete a programming assignment in Python, which includes and joins the elements specified under learning outcomes, especially the application of regular expressions.
- Written (or online) final exam for the course covers all the course material and requires a score of 50% to pass.
- A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[9] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1.5	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

Learning outcomes can be assessed through a written or oral exam, online assessments under controlled

[9] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

conditions or practical assessment (on a computer or by completing individual assignments):

- Written and/or online assessment of the knowledge of basic components and operating principles of a computer system: hardware and numerical representations. (I1)
- Written and/or online assessment of the knowledge of information and communications technology including development trends. (I4)
- Practical assessment of the use of applications in office operations through writing and text formatting assignments, creation of business presentations and basic business data collection, processing and analysis, their application in the individual preparation of a seminar paper, including the elements of learning outcomes. (I2, I3)
- Or: Group or individual seminar paper for assessing the use of applications in office operations through writing and text formatting assignments, creation of business presentations and basic business data collection, processing and analysis. (I2, I3)
- Written and/or online assessment in relation to identifying and grouping of different software licenses. (I5)
- Written and/or online assessment consisting of recognising and grouping software and programming language properties and basics of Python programming language syntax. (I6, I7)
- Creation of a programme including practical application of regular expressions for a problem-solving task including data collection, processing and presentation in Python programming language. (I2, I7)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Learning material available in the e-learning system, together with own notes and materials from lectures and exercises.
2. Zoran Kalafatić, Antonio Pošćić, Siniša Šegvić, Julijan Šribar, Python za znatiželjne - sasvim drukčiji pogled na programiranje, Element, 2016.
3. Leo Budin, Predrag Brođanac, Zlatka Markučič, Smiljana Perić, Rješavanje problema programiranjem u Pythonu, Element, 2017.
4. G. Michael Schneider, Judith Gersting, Invitation to Computer Science; (MindTap Course List), Cengage, 8th Edition, 2018.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. C. Reynolds, P. Tymann, Principles of Computer Science, Schaum’s Outline Series, McGraw-Hill, 2008.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Nataša Hoić-Božić, PhD	
Name of the course	Multimedia Systems	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge of the process of digitalization of single media (text, graphics, audio, video) and possibilities of their integration in the web site according to the guidelines for responsive web design and using multimedia standards.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe basic concepts of multimedia and multimedia elements, as well as principles and guidelines for web design and responsive design.

I2. Analyse different types of web sites according to the given guidelines for web design.

I3. Choose appropriate HTML tags and cascading style sheet elements (CSS) when creating web sites.

I4. Compare text and hypertext and design them for multimedia presentation elements by applying the HTML standards.

I5. Compare raster graphics (bitmaps) and vector graphics for print and web and create examples by applying appropriate colour models and file formats.

I6. Record, edit and adjust video and audio for web, choosing an appropriate compression standard.

I7. Embed the created samples of digitalised multimedia records for hypertext, graphics, audio and video into HTML documents.

I8. Create and publish a web site based on a designed navigation diagram, by integrating the created individual multimedia records, all in accordance with guidelines for responsive web design and using multimedia standards.

4. Course content

- Concept of multimedia and hypermedia, historical overview, application of multimedia, multimedia computer systems.
- Basic WWW concepts (HTTP, URL, HTML) and standards (HTML5) and cascading style sheets (CSS).
- Principles of web design. Graphic design, information design, interface design and navigation design. Responsive web design.
- Embedding text into computer and formatting textual content. Typography. Concepts of hypertext

and hypertextual interface elements. Application of text and hypertext on the web.

- Graphics: types of graphics (bitmaps and vector graphics), image digitalization, colour schemes, standards and graphics compression, web graphics.
- Sound digitalization. Basic patterns of audio content records, speech content, music content. Audio compression. Application of audio on the web.
- Properties and types of video. Uploading a video to a computer. Video compression and video standards. Application of video on the web.
- Basics of the development of multimedia presentations according to the ADDIE model. Application of models to design and creation of multimedia web sites.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the LMS.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the LMS and attend f2f classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Participate in discussions on a given topic on wiki (or another tool).
- Write an individual or group paper on a given topic and present it to lecturers and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[10] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion	0.5				

[10] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Discussion (on wiki or another tool) in which students jointly analyse web sites with respect to the specific criteria (I1, I2), e.g. they analyse one well designed and one poorly designed web site according to individual web design elements (graphical design, information design, interface design and navigation design etc.).
- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts regarding multimedia and web (I3, I4, I5, I6). For example, students name the characteristics of hypertext, compare bitmaps and vector graphics, describe compression formats for audio, video, graphics etc. through multiple choice questions, fill in the blank questions and essay questions.
- Practical assessment on a computer (practical preliminary exam) in which students, based on instructions and samples provided, need to create their own samples by using an appropriate programming tool (I3, I4, I5, I6). For example, they need to create graphics, audio and video similar to the ones provided, as well as a HTML document formatted using CSS that will integrate all these elements.
- Group or individual seminar paper in the form of a multimedia web presentation and corresponding preparatory documentation according to the instructions and evaluation criteria (I7-I8) set in advance. For example, students design a web site and present it using a navigation diagram and page draft, create all multimedia records (hypertext, graphics, audio, video) according to multimedia standards and integrate them into a complete web-presentation created according to the rules of responsive web design (the lecturer evaluates the paper by using the criteria known to the students before starting the assignment).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Vaughan, T. (2014). Multimedia: Making It Work, Ninth Edition 9th Edition, Berkeley: McGraw-Hill Osborne Media.
2. Hoić-Božić, N. (2015). Multimedijski sustavi, Online skripta s predavanjima u Moodle e-kolegiju
3. Beaird, J. Načela dobrog web dizajna, Site point (Dobar plan; Zagreb), 2012.
4. Niederst Robbins, J. (2018). Learning Web Design, 5th Edition (A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics), O'Reilly Media, http://www.learningwebdesign.com/
5. Hoić-Božić, N. (2018). Uvod u web dizajn, Online skripta s predavanjima u Moodle e-kolegiju.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Osborn, T. (2018). Hello Web Design: Design Fundamentals and Shortcuts for Non-Designers
2. Appropriate software manuals

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the

Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Borna Debelić, PhD	
Name of the course	Fundamentals of Economics for IT students	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students of informatics to basic principles of economics, development paths and schools, and to acquire knowledge about categorical apparatus and basic principles of economic systems, national economies, global economy system and economic entities. In line with this general objective, the task of the course is to carefully analyse basic concepts of microeconomics and macroeconomics in order to prepare students for future courses from this field.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

- I1. Describe characteristics of market structure and parts of business environment, including the consumer aspect, as well as the influence of ICT on business operations and decision-making processes within companies.
- I2. Describe principles, components and importance of financial reporting from the aspect of modern measures of business success.
- I3. Analyse information collected from various sources and, based on the analysis conducted, identify possible business opportunities, anticipate cost levels and cost trends important for efficient business decision making processes.
- I4. Analyse the effects of measures and instruments of economic policy and understand the mechanisms of their operation and manners in which they affect business operations, as well as macroeconomic developments.
- I5. Explain the importance of business planning and managing market risks, with a focus on the role and importance of information systems in the process, and create a business plan and feasibility study for an ICT project with the application of business event and transaction records for the purpose of high-quality financial reporting.
- I6. Apply quantitative and qualitative methods for making business decisions and managing risks in the ICT sector, and describe and apply entrepreneurial strategies, with the development of possible tactics and innovative solutions for improving business processes for entrepreneurs.
- I7. Determine the role and importance of services in contemporary economies based on ICT services, and design a service development strategy and efficiently manage its development.

4. Course content

- Fundamentals and development of economic science and theory, and historical overview of the relationship between economy and science
- Basic economic principles
- Microeconomics and macroeconomics
- Relationship between economy and the state, and basics of political economy
- Law of supply and demand
- Company as a basic economic entity
- Wages and labour market
- Consumption and investments
- Basics of aggregate supply and demand
- Money and basics of commercial banking
- Economic phenomena
- Strategic and operative planning and business plans
- Management, strategic planning and service development management

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☒ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments		

7. Student responsibilities

Student responsibilities for this course are as follows:

- Class attendance. Classes are held according to the blended model, as a combination of conventional classroom lessons, individual student work and student teamwork.
- Participation in continuous assessment in the form of two preliminary exams in which students have to score at least 50%.
- Creation of case studies/seminar papers on a given topic, independently or in groups.
- Taking the final written exam at the end of the semester, in which students have to score at least 50%.

8. Monitoring[11] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	
Portfolio							

[11] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Discussion in which students analyse characteristics of market structures and their specificities and the effect of ICT on business operations of a company, as well as principles, components and importance of financial reporting (I1, I2) from the aspect of modern measures of business success.
- Continuous assessment (preliminary exam) in written form, in which students demonstrate their understanding of theoretical concepts and practical implications of using economic policy measures and instruments, management and business planning (I4, I5, I6, I7), manners in which they work and affect business operations, as well as macroeconomic developments and parts of business environment and consumer impact.
- Group or individual seminar paper/case study according to the instructions and evaluation criteria (I3) set in advance.
- Final written assessment in which students demonstrate their understanding of the application and technique of business event and transaction records for the purpose of high-quality financial reporting, and the possibilities to apply entrepreneurial strategies to improve business processes and manage services (I4, I5, I6, I7).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Samuelson, P.A., Nordhaus, D.W.: Ekonomija, MATE, Zagreb, 2009.
2. Buble, M.: Osnove menadžmenta, Sinergija-nakladništvo d.o.o., Zagreb, 2006.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Gulin, D., Tušek, B., Žager, L.: Poslovno planiranje, kontrola i analiza, Hrvatska zajednica računovođa i financijskih djelatnika, Zagreb, 2004.
2. Polovina S., Medić Đ.: Osnove ekonomije - Priručnik za studij ekonomije uz udžbenik P.Samuelson W.Nordhaus, Medinek, Zagreb, 2002.
3. Ostrom, E.: Upravljanje zajedničkim dobrima: Evolucija institucija za kolektivno djelovanje, Naklada Jesenski i Turk, Zagreb, 2006.
4. North, D. C.: Institucije, institucionalna promjena i ekonomska uspješnost, Masmedia, Zagreb, 2003.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Irena Grubica, MSc / Lucia Načinović Prskalo, PhD	
Name of the course	English Language for IT Profession	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	4
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of this course is for students to acquire the vocabulary specific for information technologies and develop linguistic skills such as listening, reading, speaking and writing in English, with a special focus on topics related to IT profession.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Use the vocabulary specific for IT and terminology characteristic for IT profession, both in speech and in writing.

I2. Listen and comprehend main ideas conferred by the speaker on the topics related to IT.

I3. Read and comprehend main ideas and meaning of texts on the topics related to IT.

I4. Write short, coherent texts on the topics related to IT.

I5. Express their opinion, give instructions, descriptions and explanations from the field of IT.

I6. Collect information in English on a topic related to IT and, based on information collected, organise the material for a presentation and present main ideas.

4. Course content

Working in IT, presenting oneself and describing a job. Computer systems, description of computer hardware and software, computer architecture, computer applications – description of tasks and processes. Operating systems – exchange of technical information. Multimedia, networks, Internet. Internet servers, websites, web pages. Communication systems. Computer support – giving instructions, descriptions and explanations related to IT profession, solving user problems. Data security – description of security solutions. Academic reading, writing, expressing and presenting of materials.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	□ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students	

	will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes and actively participate in the learning process.
- Complete short papers and written assignments and hand them in within the given deadline.
- Follow course activities within the distance learning system.
- Prepare an individual or group paper on a given topic, in a written form, and present it to the lecturer and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[12] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	0.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Active participation in the learning process – during this course students will be given short assignments, in written or oral form, for the purpose of testing their ability to use the acquired vocabulary (I1), comprehend material they have read or listened to (I2, I3), express their opinion in speech or writing, give instructions, descriptions (I4, I5) and similar, all regarding topics related to IT profession. (For example, through fill in the blank questions, true or false questions related to reading comprehension etc.).
- Homework in the course of the semester in the form of short written assignments – writing short texts on a given topic related to IT profession (I1, I3, I4, I5). (Example: writing a short essay containing the instructions for the installation of certain software).
- Group or individual seminar paper in which students will tackle a given topic related to IT profession (e.g. computer peripherals) in English, organise the content of the paper and present their paper to the lecturer and other students (I1, I4, I5, I6).
- Written or online assessment (final exam) in which students will demonstrate their knowledge of the acquired vocabulary (I1), their skills of reading comprehension (I3), written expression (I4) and accuracy and precision when giving instructions, explanations and descriptions regarding topics related to IT profession (I5) (For example, through multiple choice questions, fill in the blank questions, essay questions etc.).

10. Mandatory literature (at the time of submission of study programme proposal)

[12] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

1. Glendinning, E. H., McEwan, J.: Oxford English for Information Technology 2nd Edition, Oxford University Press, Oxford, 2014.
2. Hill, D.: English for Information Technology 2, Vocational English, course book, Pearson Longman, 2012.
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Kiš, M.: Informatički rječnik : englesko-hrvatski - hrvatsko-engleski, Ljevak, Zagreb, 2000.
2. Eastwood, J.: Oxford Practice Grammar, Oxford University Press, Oxford, 2003.
3. Oxford Advanced Learner's Dictionary, Oxford University Press, Oxford, 2004.
4. Esteras, S. R.: Infotech, English for computer users, Cambridge University Press, Cambridge, 2004.
5. Powel, M.: Presenting in English, LTP Business, London, 1996.
6. Jordan, R.R.: Academic Writing Course, Study Skills in English, (7th ed.), Longman 2004.
7. McCarthy M.: O'Dell F.: Academic Vocabulary in Use; Cambridge University Press 2008.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Ana Jurasić, PhD	
Name of the course	Mathematics 2	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to basic concepts of mathematical analysis (e.g. real functions of a real variable, sequences and series, limit of a sequence, limit of a function, continuity of a function) and to basic concepts in the field of vector spaces (e.g. vector calculus, linear dependence and linear independence of vectors, linear operator). Furthermore, the objective is to encourage logical reasoning and to teach students to apply mathematical knowledge when solving problems from the field of information sciences.

2. Course enrolment requirements

Previously taken course Mathematics 1.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Define elementary real functions of a real variable, name their properties and sketch their diagrams.

I2. Use elementary functions and their properties when solving simple real problems.

I3. Solve a standard problem from the field of sequences, limits of sequences and series.

I4. Apply sequences and series when solving standard problems from the field of informatics.

I5. Apply the limit of a function when testing the continuity of a function.

I6. Express main terms, definitions and theorems and explain concepts from the field of vector spaces and linear operators.

I7. Solve standard problems in mathematics and informatics by applying concepts and methods of vector calculus and analytic geometry in three-dimensional space.

4. Course content

- Basic concepts and classification of real functions of a real variable.
- Properties of real functions of a real variable.
- Elementary functions and their role in application.
- Concept and properties of real number sequences. Accumulation point and limit of a sequence. Properties of limit of a sequence. Concept of series. Geometric series. Series convergence criteria.
- Limit of a function. Properties of limit of a function. Continuity of a function.
- Concept of vector space. Basis and dimension of a vector space.

- Linear operators. Basic properties of linear operators.
- Vector calculus and analytic geometry in three-dimensional space.
- Application of linear algebra to problems from the field of information sciences.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Lectures are delivered in classroom, and exercises are partly auditory and partly on computers. E-learning system is also used. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes, participate in all course activities and follow notifications related to classes that can be found within the e-learning system.
- Participate in continuous assessment (preliminary exams) and achieve the score anticipated in the course syllabus.
- Complete (individually or in teams) a given problem-solving task and achieve the score anticipated in the course syllabus.
- Score at least 50% on the final exam.

All the other details will be provided in the course syllabus.

8. Monitoring[13] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	2.5	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- At the written or oral exam, students are required to define basic concepts and name characteristic examples from the field of real functions of a real variable, sequences and series, limit of a sequence and limit of a function, and continuity of a function. For example, define a concept of real number sequence and provide an example. (I1, I3, I5)
- At the written or oral exam, students are required to express and explain basic theorems, and to derive simple formulas from the field of real functions of a real variable, sequences and series, limit of a sequence and limit of a function, and continuity of a function. For example, express a theorem

[13] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

related to the properties of limit of a function. (I1, I3, I5)

- At the (written) preliminary exam, students are required to define certain elements of a function, connect functions and their diagrams and name their properties. (I1)
- At the (written) preliminary exam or within a project assignment, students are required to model simple problems using the properties of elementary functions (quadratic, exponential, trigonometric etc.). For example, determine the maximum level of computer sales within a certain period and the total earnings after a given time. (I2)
- At the (written) preliminary exam, students are required to determine the limit of a given sequence or the sum of a given sequence. (I3)
- At the (written) preliminary exam or within a project assignment, students are required to model simple problems using the properties of sequences and series. (I4)
- At the (written) preliminary exam, students are required to analyse the continuity of a given function. (I5)
- At the oral or written exam or through assignments implemented for e-learning self-assessment, students are required to define basic concepts from the field of vector spaces and linear operators, and to provide typical examples. For example, define vector space and subspace. (I6)
- At the oral or written exam or through assignments implemented for e-learning self-assessment, students are required to express and explain basic theorems from the field of vector spaces and linear operators, and to provide typical examples. For example, determine the kernel, image, rank and defect of a linear operator. (I6)
- At the (written) preliminary exam, students are required to determine linear dependence and linear independence of a set of vectors in different vector spaces, to construct different bases and to link vector views in different bases of the same vector space. (I7)
- Students are required to independently complete a problem-solving task from mathematics, graphics or related field by using appropriate software (for example, use of three-dimensional graphic sketch, transformation in space or computer animation). The solution is handed in in the form of a written mathematical text accompanied by a programming solution in an appropriate programming tool. (I7)

10. Mandatory literature (at the time of submission of study programme proposal)

1. B. Divjak, T. Hunjak, Matematika za informatičare, TIVA, Fakultet organizacije i informatike, Varaždin, 2004.
2. P. Javor, Uvod u matematičku analizu, Školska knjiga, Zagreb, 1992.
3. A. Agljić Aljinović, N. Elezović, D. Žubrinić, Linearna algebra, Element, Zagreb, 2011.
4. D. Bakić, Linearna algebra, Školska knjiga, Zagreb, 2008.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. B. Divjak, T. Hunjak, Zbirka zadataka iz matematike, TIVA, Fakultet organizacije i informatike, Varaždin, 2002.
2. P. Javor, Matematička analiza: Zbirka zadataka; teoremi i definicije, riješeni zadaci, Školska knjiga, Zagreb 1990.
3. Demidovič, Zadaci i riješeni primjeri iz više matematike, Tehnička knjiga, Zagreb.
4. V. P. Minorski, Zbirka zadataka više matematike, Tehnička knjiga, Zagreb, 1971.
5. A. Agljić Aljjinović, N. Elezović, Linearna algebra, zbirka zadataka, Element, Zagreb, 1995.

6. K. Horvatić, Linearna algebra, Golden marketing – Tehnička knjiga, Zagreb, 2004.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Maja Matetić, PhD	
Name of the course	Programming 2	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about basic data structures (linked list, circular list, queue, stack, binary search tree) and algorithms (searching and sorting), as well as advanced programming techniques (divide and conquer, recursion, dynamic programming). The objective of the course is to teach students how to develop complex and sophisticated programmes.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

- I1. Build own subroutine libraries with commonly used subroutines and include them into programs if necessary.
- I2. Develop a recursive solution to a given problem-solving task and execute it using a chosen imperative programming language.
- I3. Choose an appropriate implementation of linear (linked list, circular list) abstract data type for a given problem-solving task and implement it.
- I4. Choose an appropriate implementation of linear (stack and queue) abstract data type for a given problem-solving task and implement it.
- I5. Choose an appropriate implementation of tree (e.g. binary search tree) abstract data type for a given problem-solving task and implement it.
- I6. Determine time and space complexity of operations for a given implementation of abstract data type.

4. Course content

Sorting algorithms. Searching algorithms. Introduction to pointers. Dynamic memory allocation. Pointers and dynamic fields. Pointers and linked lists. Doubly linked lists. Circular list, doubly linked lists. Stack and queue. Trees. Recursion. Dynamic programming. “Divide and conquer” technique. Chosen algorithms.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning	

	management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system, prepare for classes (do homework?) and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical quizzes and preliminary exams) and successfully pass them.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[14] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Preliminary exams	1.5	Quizzes	1		

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Online theoretical assessment (quiz) to test the understanding of theoretical concepts related to data structures such as linked lists, circular lists, queues, stacks and binary search trees and programming techniques (I1, I2, I4).
- Practical assessment (practical preliminary exam) in which students need to choose an appropriate implementation of linear abstract data type for a given program solving task, e.g. a stack for inverse printout of input values, conversion of infix to prefix notation of arithmetic expressions (I3).
- Practical assessment (practical preliminary exam) in which students need to design an algorithm for solving a given problem and implement it for a given data structure, e.g. for a binary search tree (I5).
- Practical assessment (homework, preliminary exam) in which students analyse time and space complexity of operations for a given implementation of abstract data type, e.g. for different searching and sorting algorithms derived on the list (I6).
- Practical assessment (final exam) in which students apply the acquired programming skills and knowledge to implement more complex algorithms according to the set instructions and evaluation criteria (I2, I3, I4, I5).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Julijan Šribar, Boris Motik: Demistificirani C++, Dobro upoznajte protivnika da biste njime ovladali, Element, Zagreb, 2001.

[14] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

2. Maja Matetić: Skripta uz predmet Programiranje 1 (digitalna skripta), Odjel za informatiku, Sveučilište u Rijeci, Rijeka 2012.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Algorithms in C, Parts 1-4, Fundamentals, Data structures, Sorting, Searching, Robert Sedgewick, Addison-Wesley, 1998
2. Vulin, R.: Zbirka riješenih zadataka iz C-a, Školska knjiga, Zgb, 2003.
3. Walter Savitch: Problem Solving in C++, Pearson Publishing, 2006.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Božidar Kovačić, PhD	
Name of the course	Operating Systems	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about operating systems and processes within operating systems, about basic concepts related to operating systems – process, process adjustment mechanisms, data management, memory management, and to acquire knowledge and skills for advanced use of operating systems.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Name types of operating systems and explain basic tasks of operating systems with respect to their structure.

I2. Interpret program execution with the application of processes and threads, and link processes and threads to execution states.

I3. Analyse mutual exclusion mechanisms for processes and threads, and apply appropriate mutual exclusion mechanisms to problem solving, i.e. coordinating the simultaneous operation of processes and threads.

I4. Analyse memory management strategies and choose an appropriate memory management strategy for a specific problem-solving task.

I5. Analyse types of file systems and choose an appropriate file system according to the given system specifications.

I6. Link parts of operating system and hardware used for managing input-output units.

I7. Analyse security mechanisms in operating systems and justify the use of basic and additional protection functionalities for a specific operating system.

4. Course content

- Introduction to operating systems: development of operating systems, basic tasks of operating systems, structure of operating systems.
- Interaction (relationship) between an operating system and hardware, process management: process concurrency, synchronisation, delays, CPU management.
- Memory management: paging, segmentation, placement strategies, memory protection.

- Resource allocation, data management: working with files and directories.
- Managing input-output devices: driver, controller, interrupt processing.
- Role of security and protection in operating systems: security mechanisms, protection implementation in the operation of processes and threads.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). A detailed schedule with lectures and exercises will be defined in the syllabus. When they enrol into this course, students will be instructed to use the distance learning system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Write an individual or group paper on a given topic and present it to lecturers and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[15] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- In the written exam, students name types of operating systems, sketch parts and connections of operating systems and provide explanations of basic tasks of operating systems (I1).
- In the written exam, students write a program consisting of several processes and threads, explaining the results of program execution (I2).
- As their homework, students design a program that correctly coordinates a given system consisting

[15] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

of several processes and threads by applying an appropriate mutual exclusion mechanism (I3).

- In the written exam, students complete a problem-solving task, in which they have a given memory management strategy, with certain limitations (I4).
- In the written or oral exam, students choose a file system and sketch the description of file placement on the drive (I5).
- In the written exam, students identify the activity of the operating system and hardware during the processing with input-output units (I6).
- In the written exam, students identify security and protection mechanisms of a given operating system (I7).
- Students write a group or individual seminar paper in which they analyse individual concepts of operating systems e.g. ones related to memory management, file systems, security mechanisms etc. according to the instructions and evaluation criteria set in advance. (I4-I7)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Tanenbaum A., Modern Operating systems, Pearson, 2014.
2. Silberschatz, A., P.B. Galvin, G. Gagne, Operating System Concepts, 9th edition, John Wiley&Sons, New York, 2012.
3. Operacijski sustavi. Budin, L., Golub, M., Jakobović, D., Jelenković L. Element, Zagreb, 2010.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Love R., Linux Kernel Development (3rd Edition), Addison-Wesley 2010.
2. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Mile Pavlić, PhD	
Name of the course	Data Modelling	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to teach students how to analyse business documentation and interview users, and how to create data models and logical schemas of relational databases.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Compare different methodologies and tools for conceptual data modelling.

I2. Analyse business documents of a company, independently and/or by interviewing business users and document it according to specific criteria.

I3. Create a conceptual data model.

I4. Revise a conceptual data model and create a logical data model using basic principles of normalization process.

I5. Master basic concepts of organization and apply methods and techniques to the design of organisation and adjust information system.

4. Course content

- Information system design, methods and tools for data modelling, MIRIS - Information Systems Development Methodology, project realization
- abstraction
- entity-relationship method, entity-relationship diagram (ERD), entities, relationships, attributes, cardinality, candidate for entity type key
- limitations over data model
- translation of ERD into relational data model
- basic principles of normalization
- metamodelling
- basic concepts of organization theory
- methods and techniques of organization design
- coordination of information and organization systems

- analysis of organization system documentation data and content

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held by combining classroom work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment and achieve the number of credits equal to or higher than the passing score (if any).
- Participate in practical problem-solving tasks and achieve the number of credits equal to or higher than the passing score (if any).
- Individually or in teams, make a project and present it to the lecturer, and achieve the number of credits equal to or higher than the passing score (if any).
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[16] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam	1	Essay		Research	
Project	1.5	Continuous assessment		Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts from the field of data modelling (I1, I3, I4, I5), e.g. students list the characteristics of different methodologies for conceptual data modelling through multiple choice questions, fill in the blank questions and essay questions.
- Practical problem-solving task in which students need to design a data model by applying appropriate methods and tools (I3, I4). For example, design a data model for the document "Travel

[16] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Order" using entity-relationship method and revise it using normalization principles.

- A project in which students, individually or in groups, need to choose methods for the collection of user requirements, create models and project documentation. Documentation needs to contain entity-relationship diagrams and a relational database schema (I2, I3, I4). Students present their solution to the lecturer. For example, for a chosen system (e.g. Student Service), students need to collect user requirements, analyse the documentation, create entity-relationship diagrams, translate them to relational database model and present their solutions to the lecturer.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Scripts, presentations and other learning material available in the e-course.
2. Pavlić, M., Oblikovanje baza podataka, Odjel za informatiku, Sveučilište u Rijeci, Rijeka, 2011.
3. Pavlić, M., Informacijski sustavi, Školska knjiga, Zagreb, 2011.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Valacich J. S., George J. F Modern Systems Analysis and Design. 8th ed. Pearson Education, Inc, 2017.
2. Pavlić, M., Jakupović, A., Čandrlić, S. Modeliranje procesa, Odjel za informatiku, Sveučilište u Rijeci, Rijeka, 2014.
3. Batini, C., Ceri, S., Navathe, SB., Conceptual Database Design: An Entity-relationship Approach, Benjamin/Cummings Publishing Company, 1992.
4. Elmasri, R., Navathe, S., Fundamentals of database systems. Addison-Wesley Publishing Company, 2010.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Ivo Ipšić, PhD	
Name of the course	Computer Architecture and Organization	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to the basics of computer system organization and to the basic concepts of computer system operation.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse the CPU mode of operation and instruction set.
I2. Analyse principles of operation of different RISC and CISC processor architectures.
I3. Evaluate computer performance and effects of computer architecture on its performance.
I4. Critically argue about the proposed optimal configuration with respect to its performance and price.
I5. Choose a software solution for efficient execution of the instruction set.
I6. Adjust the software solution to the characteristics of functional components of the computer.
I7. Write simple programmes using an assembler.

4. Course content

Classification of computer architecture. Von Neumann computer model. Structure of a simple microprocessor: control unit, arithmetic logic unit. Execution of instructions of a simplified microprocessor model. Microprogrammed and hardwired control unit. MIPS processor pipeline architecture. Memory systems. Cache. Virtual memory. Computer performance analysis. Input-output computer systems. Interrupt and exception handling. Multicore processors and graphics processing units. Examples of assemblers for 32 and 64-bit microprocessors.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes, participate in all course activities and follow course activities within the distance learning system.
- Participate in continuous assessment (theoretical and practical preliminary exams and homework).
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[17] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1.5	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Online assessment (homework) in which students, e.g. through multiple choice questions, fill in the blank questions and essay questions, list the characteristics of RISC and CISC processor architecture, describe the process of executing certain instructions in the microprocessor and analyse the status of the microprocessor during the execution. Through essay questions and fill in the blank questions, students demonstrate their knowledge of basic concepts and standards related to performance measurements for different computer subsystems. (I1-I3)
- Practical assessment (online preliminary exam), in which students complete problem-solving tasks, demonstrating the use of synthetic tests and analysing the results obtained, based on which they identify key architecture components that affect test results and potential performance bottlenecks. Based on the results, they propose potential alternatives and argue about their justification in relation to the expected increase in performance with respect to price. (I3, I4)
- Assessment (preliminary exam) in which students complete problem-solving and programming tasks using an assembler. For example, they analyse a programme segment consisting of several instruction sets and determine the state of the processor (registers, stack etc.) during the execution of such instructions. By analysing the execution of given instructions, they propose time sequence of the instructions that will enable efficient execution of such instructions. (I5, I6)
- Assessment (final exam) in which students complete problem-solving and programming tasks using an assembler, e.g. they write a simple program using an assembler, demonstrating they know how to use input-output devices. (I7)

10. Mandatory literature (at the time of submission of study programme proposal)

1. S. Ribarić. "Građa računala", Algebra d.o.o., Zagreb, 2011.
2. J. L. Hennessy, D. A. Patterson. "Computer Organization and Design MIPS Edition: The Hardware/Software Interface", 5th edition, Morgan Kaufmann Pub., San Mateo, 2014.

[17] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

3. Scripts, presentations and other learning material available in the e-course.		
11. Optional/additional literature (at the time of submission of the study programme proposal)		
12. Number of assigned reading copies in relation to the number of students currently attending the course		
Title	Number of copies	Number of students
13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences		
Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).		

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Davor Dragičević, PhD	
Name of the course	Fundamentals of Probability and Statistics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	1	
ECTS credits and manner of instruction	ECTS credits	4
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master basic terminology and results of probability theory and statistics, and apply them to problem-solving tasks.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Calculate the probability of an event by applying the fundamentals of combinatorics, properties of probability and Venn diagrams.

I2. Apply Bayes’ rule and conditional probability tree to calculate conditional probabilities.

I3. Evaluate the probability of a continuous random variable to assume a value higher (or lower) than a given number using the distribution table or by applying statistical software.

I4. Draw a density function chart of the most commonly used random variables (standard normal distribution, student’s distribution, χ^2 distribution).

I5. Identify limitations of various data collection methods and other sources of bias, and recognise data types with respect to measurement scale.

I6. Apply graphical and numerical methods of descriptive statistics using an appropriate statistical software (e.g. R, SPSS, Statistica, SAS etc.).

I7. Apply parameter estimation and hypothesis testing methods to analyse one variable or correlation between two variables by using an appropriate statistical software for the purpose of understanding social or natural phenomena or making data-driven decisions.

I8. Interpret the results of statistical data analysis in the context of specific questions, using the vocabulary appropriate for the context of the task, and evaluate the validity of data-driven statements.

4. Course content

Fundamentals of combinatorics. Probability space. Laplacian model. Conditional probability. Independence. Formula of total probability and Bayes’ formula. Geometric probability. Random variables. Mathematical expectation and variance. Density function and distribution function. Continuous random variables. Normal distribution. Descriptive statistics. Mean values. Measures of dispersion. Measures of asymmetry and roundness. Parameter estimation. Credible intervals. Hypothesis testing.

5. Manner of instruction	☒ lectures	☒ individual assignments

	□ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. Course exercises consist of auditory exercises (in classroom) and computer exercises. A detailed schedule will be provided in the course syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly participate in all course activities.
- Participate in continuous assessment (preliminary exams and computer assessment).
- Prepare an individual or group seminar paper on a given topic, in a written form, and present it to lecturers and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[18] of student work

Class attendance	1	Class participation	0.5	Seminar paper	0.5	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written assessment (preliminary exams) in which students apply terminology and results of probability theory to problem-solving and computational tasks (I1, I2, I3, I4), e.g. they solve specific computational and problem-solving tasks on paper (such as the following one: “Calculate the probability of getting a sum greater than 10 when two dice are rolled”).
- Practical assessment on a computer (computer assessment) in which students, based on instructions and given examples need to perform statistical data analysis using an appropriate software tool (I6, I7, I8), e.g. calculate point and interval estimates for a given data set and test certain statistical hypotheses.
- Group or individual seminar paper in the form of a web presentation and accompanying preparatory documentation prepared according to the instructions and evaluation criteria set in advance (I5, I6, I7, I8), e.g. students choose and analyse a real data set using a chosen statistical software, and then produce and present a report on performed statistical data analysis (the lecturer evaluates the paper

[18] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

by using the criteria known to students before starting the assignment).

- Final exam (written and/or oral) in which students demonstrate their understanding of theoretical concepts of probability theory and statistics (I1, I2, I3, I4, I5), e.g. through multiple choice questions and essay questions they enumerate basic properties of probability, provide examples of independent and dependent events, definitions of conditional probability and mathematical expectation of a random variable...

10. Mandatory literature (at the time of submission of study programme proposal)

1. N. Sarapa: Vjerojatnost i statistika, I i II dio, Školska knjiga, Zagreb, 1993.
2. I. Šošić: Primijenjena statistika. 2. izmijenjeno izd., Školska knjiga, Zagreb, 2006.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. N. Sarapa: Teorija vjerojatnosti, Školska knjiga, Zagreb, 2002.
2. K. Kero, J. Dobša, B. Bojanić-Glavica: Statistika deskriptivna i inferencijalna i vjerojatnost, Tiskara Varteks, Varaždin, 2008.
3. T. Pogány: Teorija vjerojatnosti – Zbirka riješenih ispitnih zadataka, Odjel za pomorstvo Sveučilišta u Rijeci, Rijeka,1999.
4. M. Papić: Primijenjena statistika u MS Excelu, Zoro, Zagreb, 2012.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Marija Maksimović, PhD	
Name of the course	Mathematics 3	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master basic terminology and results of differential and integral calculus for functions of single or multiple variables, and apply them to problem-solving tasks.

2. Course enrolment requirements

A pass mark in Mathematics 2.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain the concept of derivative of a real function of a real variable and geometric interpretation of derivative at a point.
I2. Analyse the flow of elementary function using derivatives and sketch its diagram.
I3. Apply differential calculus for finding local extrema of a function of a single variable and inflection points of a function.
I4. Determine a primitive function and apply integral calculus for the calculation of surface area and volume.
I5. Explain the concept of derivative of a function of several variables and geometric interpretation of partial derivative.
I6. Analyse elementary functions of two variables by applying differential calculus.
I7. Determine local and conditional extrema of a function of several variables.
I8. Analyse and complete a problem-solving task from the field of mathematical analysis of a function of single or multiple variables with the application of appropriate software tool or own software solution, and present the solution in writing as a correct mathematical text.

4. Course content

Concept of derivative of a function of single real variable. Derivation rules. Higher-order derivatives. Derivation of a parametrically defined function. L'Hospital's rule. Basic theorems of a differential calculus. Monotony intervals and extrema of a function. Concavity and convexity of a function. Inflection points. Asymptote of a function. Flow of a function. Primitive function and indefinite integral. Basic methods of integration. Definite integral. Calculation of definite integral. Application of integrals to calculation of surface areas and volumes. Concept of derivative of a function of several variables. Differential and partial derivatives. Gradient. Tangent plane. Local and conditional extrema.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories

	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. Course exercises consist of auditory exercises (in classroom) and computer exercises. A detailed schedule will be provided in the course syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly participate in all course activities.
- Participate in continuous assessment (preliminary exams and tests).
- Write a seminar paper on a given topic.
- Score at least 50% on the final exam.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[19] of student work

Class attendance	1	Class participation	0.5	Seminar paper	0.5	Experimental work	
Written exam	0.5	Oral exam	1	Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written assessment (preliminary exams and tests) in which students apply terminology and results of mathematical analysis of functions of single or multiple variables to problem-solving and computational tasks (I2, I3, I4, I6, I7), e.g. they solve specific computational and problem-solving tasks on paper (such as the following one: “For the given function, determine a domain, zero, monotony intervals, local extrema, inflection points and asymptotes and sketch its diagram”).
- Individual or group seminar paper to present the solution to a problem-solving task from the field of mathematical analysis of a function of single or multiple variables, with an addendum related to own software solution or the use of appropriate software tool (I8). For example, students model and solve a real problem using mathematical analysis of a function of single or multiple variables, and then present it in writing (the lecturer evaluates the paper by using the criteria known to students before starting the assignment).
- Final exam (written and/or oral) in which students demonstrate their understanding of theoretical concepts related to differential and integral calculus (I1, I2, I3, I4, I5, I6, I7), e.g. through essay questions they explain geometric interpretation of derivative at a point, they cite derivation and integration rule, analyse the flow of a function etc.

[19] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

10. Mandatory literature (at the time of submission of study programme proposal)

1. B. Divjak, T. Hunjak: Matematika za informatičare. TIVA, Fakultet organizacije i informatike, Varaždin, 2004.
2. B. Divjak, T. Hunjak: Zbirka zadataka iz matematike, TIVA, Fakultet organizacije i informatike, Varaždin, 2002.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. P. Javor: Uvod u matematičku analizu, Školska knjiga, Zagreb, 1992.
2. P. Javor: Matematička analiza: Zbirka zadataka; teoremi i definicije, riješeni zadaci, Školska knjiga, Zagreb 1990.
3. B. P. Demidovič: Zadaci i riješeni primjeri iz više matematike, Tehnička knjiga, Zagreb.
4. S. Kurepa: Matematička analiza III, Tehnička knjiga , Zagreb (više izdanja).

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Mile Pavlić, PhD	
Name of the course	Business Process Analysis	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to teach students how to independently perform analyses, interview users, collect user requirements and design process models, as well as to develop the designer's mindset, with a high level of critical attitude towards the results of the analysis and models obtained.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Choose an appropriate method for designing a process model.

I2. Determine processes and business documentation flow within a business system.

I3. Analyse methods and techniques and design a procedure for collecting user requirements in order to develop an information system.

I4. Demonstrate UML modelling.

I5. Design a process model.

I6. Evaluate designed process models.

I7. Link user requirements, business process models and data models for a given business system.

4. Course content

- Designing a process model, process modelling methods, stages and activities of a process model development life-cycle, MIRIS – Information Systems Development Methodology;
- Structured system analysis, business functions, business processes, existing and future system statuses, feasibility, costs and benefits; interviewing, structure testing;
- Data flow diagram, process, types of processes, data flow, data base, external system;
- Decomposition, system context, hierarchical system description; Limitations of process model, law of conservation of data flows, decomposition criteria; drawing recommendations;
- Process for designing a process model;
- Means for representing process logic; means for representing data warehouse structure;
- Main project, project task, team analysis;
- Methods: SSA, DFD, activity flow diagram, decision tree, Nassi–Shneiderman diagram, decision

tables, Warnier-Orr diagram;

- How to develop an IS in a company.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment and successfully pass it.
- Undertake practical problem-solving tasks and successfully complete them.
- Make a project, independently or in groups, and present it to the lecturer.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[20] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	2	Continuous assessment		Report		Practical work	0.5
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts from the field of process analysis and modelling (I1, I2, I3, I4) E.g. through multiple choice questions, fill in the blank questions and essay questions, students list the characteristics, advantages and disadvantages of different models and techniques for collecting user requirements.
- A project in which students, individually or in groups, need to choose methods for the collection of user requests, create models and project documentation. Such documentation should contain key process description parameters and definitions of data sets used by the processes (I2, I3, I5, I6). Students present their solution to the lecturer.
- Practical problem-solving task in which students need to create a business model that includes both

[20] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

the process and the data component by applying appropriate methods and tools (I7).

- Practical problem-solving task in which students need to model user requirements using an UML technique (I4), for example create an activity diagram.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Pavlić, M., Jakupović, A., Čandrlić, S. Modeliranje procesa, Odjel za informatiku, Sveučilište u Rijeci, Rijeka, 2014.
2. Fowler, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition), Pearson Education, Boston, 2004.
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Brumec, J., Brumec, S. Modeliranje poslovnih procesa, Redak, Split, 2016.
2. Freund, J., Rücker, B. Real-Life BPMN, 2016.
3. Daoust, N., UML Requirements Modeling For Business Analysts, Technics Publications, Westfields, 2012.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Marina Ivašić-Kos, PhD	
Name of the course	Object-Oriented Programming	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about object-oriented paradigm and learn how to apply standard concepts of object-oriented paradigm in system modelling and implementation using a chosen object-oriented programming language.

The objective is to teach students how to independently analyse and specify requests, develop models and programs by using object-oriented approach to solve problem-solving tasks.

2. Course enrolment requirements

Previously taken courses Programming 1 and Programming 2.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain concepts related to object-oriented paradigm such as class, object, data privacy and encapsulation, constructors and destructors, relationships between classes, class hierarchy, abstraction, inheritance, polymorphism.

I2. Design and model basic class concepts such as constructors, member attributes and methods with defined visibility and present them using an appropriate diagram (class diagram, activity diagram or sequence diagram).

I3. Implement a class with concepts such as constructors, member attributes and methods using an appropriate programming language.

I4. Design and model concepts of an object-oriented model such as encapsulation, relations of association and class hierarchy, inheritance, overloading, polymorphism and present them using a class diagram.

I5. Implement concepts of an object-oriented model such as association and class hierarchy, inheritance, overloading, overriding and polymorphism using an appropriate programming language and based on a designed class diagram.

I6. Compare and analyse different model implementations in the object-oriented paradigm such as the use of standard operators, friends of a class and class methods.

I7. Apply skills and knowledge from object-oriented paradigm to solve problem-solving tasks.

4. Course content

- Introduction to object-oriented modelling and programming. Standards and specificities of a chosen object-oriented language (C++). Concepts related to object-oriented paradigm such as class, object, data privacy and encapsulation, constructors and destructors, relationships between classes, class hierarchy, abstraction, inheritance, overloading, polymorphism.

- Modelling of basic class concepts such as constructors, member attributes and methods with defined visibility by using UML structure diagrams (class diagrams, object diagrams).
- Definition of classes with member attributes and functions with defined visibility. Constructors and destructors. Overloading of constructors and functions. Use of basic system classes and functions, and user-defined classes. Dynamic class definition. Copy constructor and class references. Complex classes, class strings, vectors.
- Modelling changes in object state (activity diagram, statechart diagram) and object interaction (sequence diagram, communication diagram).
- Relations between classes. Inheritance: types and application of inheritance. Modelling and implementation of inheritance. Class hierarchy and multiple inheritance. Function overriding and function overloading. Abstract classes, polymorphism, virtual classes.
- Function and class templates. Operator overloading. Selected chapters from STL library.
- Examples and analysis of object-oriented models and implementation of solutions to problem-solving tasks from various fields of application.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams).
- Design, create and present a solution to a problem-solving task (individually or in pairs) and score at least 50% on the final exam.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[21] of student work

[21] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts of object-oriented paradigm and analyse different model implementations. (I1, I6)
- Practical assessment in which students, using UML diagram tools, create class diagrams containing concepts such as class hierarchy, association, inheritance, overloading, polymorphism and interaction diagrams or activity diagrams, according to the given specification and problem-solving task. (I2, I4)
- Practical assessment (practical preliminary exam) in which students, on a computer and using a given programming language, implement a given class diagram and interaction diagram, with concepts such as class hierarchy, abstraction, inheritance and associations between classes, overloading, polymorphism. (I3, I5)
- Final exam: Practical project assignment in which students apply their skills and knowledge from the field of object-oriented paradigm to solve problem-solving tasks related to the topic of their choice, according to the instructions and evaluation criteria set in advance. (I7)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Robert Lafore: Object-Oriented Programming in C++ (4th Edition), eBook, pdf, 2001.
2. Bjarne Stroustrup: The C++ Programming Language, 4th Edition, Addison-Wesley; 2013, pdf.
3. B. Stroustrup: Programming -- Principles and Practice Using C++ (Second Edition), Addison-Wesley, 2014.
4. Grady Booch: Object-Oriented Analysis and Design with Applications (3rd Edition), 2007, pdf.
5. M. Ivašić-Kos: Objektno progamiranje – C++, on-line prezentacije predavanja, zadaci i primjeri riješenih zadataka, Moodle e-knjiga, 2018.
6. M. Ivašić-Kos: Objektno modeliranje – UML, on-line prezentacije predavanja, zadaci i modeli različitih problemskih situacija, Moodle e-knjiga, 2018.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Tony Gaddis: Starting Out with C++ from Control Structures to Objects (9th Edition), 2017.
2. Erich Gamma: Design Patterns: Elements of Reusable Object-Oriented Software, 2009, pdf.
3. Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, 2015.
4. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14, Scott Meyers, 2014.
5. B. Lippman: C++ Primer (5th Edition), Stanley, 2013, pdf.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences		
Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).		

COURSE DESCRIPTION		
Course instructor	Full Prof. Mario Radovan, PhD / Vedran Miletić, PhD	
Name of the course	Computer Networks	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master basic knowledge about computer networks, Internet, web applications and protocols, as well as to acquire skills for using such knowledge.

2. Course enrolment requirements

A pass mark in Fundamentals of Informatics.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Classify and compare reference models of computer system network architecture and name the role of each layer within these reference models.

I2. Explain the method of operation of chosen services and protocols from individual layers of reference network architecture models.

I3. Analyse major Internet protocols using protocol documentation and software tools.

I4. Name challenges in the domain of computer network security and describe solutions for responding to such challenges.

I5. Apply protocols of Internet application layer using the appropriate software tools.

I6. Recognise and express development trends in the information and communications technology in the domain of computer networks.

4. Course content

- Basic terminology related to computer networks and the Internet. Network edge and network core. Basic properties of networks. History of computer networking and Internet development.
- Application layer. Web. Electronic mail. Domain Name System. Peer-to-peer applications. Web application development.
- Transport layer. Multiplexing and demultiplexing. Connectionless data transfer. Reliable data transfer. Connection-oriented data transfer. Congestion management.
- Network layer. Virtual circuit and datagram. Router. Packet forwarding and Internet addressing. Routing. Broadcast and multicast.
- Data link layer. Troubleshooting. Multiple access links and protocols. Switches and local area networks.
- Wireless and mobile networks. Wireless connections. Wireless local area networks. Internet access through mobile networks. Mobility.

5. Manner of instruction	☒ lectures	☒ individual assignments

	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Complete problem-solving tasks as part of homework and hand in the solutions within the deadline.
- Answer questions and complete problem-solving tasks during auditory and/or laboratory exercises.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[22] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	1
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- As part of their homework, in the form of online assessment, students are required to hand in their files with solutions to problem-solving tasks, demonstrating their ability to set up a network model and/or required network service configuration by using a network simulator, network emulator and server and client applications (I3, I5).
- In laboratory exercises, students are required to take written or online assessment, in which they demonstrate their understanding of theoretical concepts necessary to complete programming tasks, e.g. through multiple choice questions, fill in the blank questions and essay questions (I2, I4). Following the written or online assessment, students are required to solve problem-solving tasks and hand in the files with solutions through online assessment, thus demonstrating their ability to set up a network model and/or required network service configuration by using a network simulator, network emulator and server and client applications (I3, I5).
- Written or online assessment in which students demonstrate their understanding of theoretical concepts related to computer networks and the Internet, e.g. through multiple choice questions, fill in the blank questions and essay questions (I1, I2, I4, I6).

[22] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Kurose, J. F. & Ross, K. W. Computer networking: a top-down approach. (Pearson, 2013).
2. Peterson, L. L. & Davie, B. S. Computer networks: a systems approach. (Morgan Kaufmann, 2012).
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Bažant, A., Gledec, G., Ilić, Ž., Ježić, G., Kos, M., Kunštić, M., Lovrek, I., Matijašević, M., Mikac, B. & Sinković, V. Osnovne arhitekture mreža. (Element, 2014).
2. Halsall, F. Computer networking and the Internet. (Addison-Wesley, 2006).
3. Tanenbaum, A. S. & Wetherall, D. Computer networks. (Pearson/Prentice Hall, 2011).
4. Sterbenz, J. P. G. & Touch, J. D. High speed networking: a systematic approach to high-bandwidth low-latency communication. (Wiley, 2001).
5. Comer, D. Computer networks and Internets. (Pearson, 2015).
6. Comer, D. Internetworking with TCP/IP. (Pearson/Prentice Hall, 2013).

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Patrizia Poščić, PhD	
Name of the course	Databases	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master basic knowledge about databases, with a focus on relational databases. This knowledge includes, but is not limited to, logical database design, relational algebra and non-procedural query language (SQL).

2. Course enrolment requirements

Previously taken course Mathematics 1.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain basic terminology related to the database theory, as well as concepts of relational data model.

I2. Compare query execution methods by using a theoretical query language and a database query language.

I3. By applying logical database design methods, create or modify a given logical model, thus removing anomalies from databases.

I4. Set (design) a development environment for a selected database management system by creating user rights and roles and ensuring a satisfactory database security level.

I5. Based on a logical model, create a database using a selected database management system, as well as its basic objects and structures (e.g. tables, views, keys).

I6. Determine the conditions of entity integrity and referential integrity in the implemented database.

I7. By using the selected query language, modify the existing database and create simple and complex queries for information within the database.

I8. Within a given programming environment, design a software solution based on a relational database.

4. Course content

Introduction to databases. Database concepts. Relational database. Relational algebra.

Operations in the relational model. Non-procedural languages for working with relational databases – SQL.

Integrity rules in relational data model. Concept of null value and incomplete information. Elements of dependency theory. Normalization; normal forms. Software for application development based on relational databases.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes, participate in all course activities and follow course activities within the distance learning system.
- Participate in continuous assessment (theoretical and practical preliminary exams and quizzes).
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[23] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practical work	
Portfolio		Discussion		Individual assignments	0.5		

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through continuous assessment (a theoretical or practical preliminary exam and quizzes) and a written exam, accompanied by computer work.

- In the theoretical preliminary exam, students demonstrate their understanding of theoretical concepts related to databases and the relational data model (I1) and they correlate and compare different query execution methods by using a theoretical query language and a database query language (I2). E.g. explain the limitations of primary key or translate a given SQL query into relational algebra and vice versa.
- At the written exam, students create an appropriate logical data model based on a predefined conceptual model and by applying appropriate rules, and identify the normal form of the existing relational schema, and perform database normalization. E.g. translate an entity-relationship diagram into relational data model or normalize a given relational schema to third normal form. (I3)

[23] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- Students independently create a development environment for the database management system (including the distribution of user privileges and a satisfactory security level) on a computer. E.g. students independently and/or according to the instructions determine user groups and privileges for working with a database and define them on a computer. (I4)
- In the practical preliminary exam and/or quiz on a computer, students create a database using the database management system based on the created logical model. E.g. students independently create objects and structures within a database (relations, attributes, keys, indices, views etc.) on a computer. (I5)
- In the practical preliminary exam and/or quiz, students determine the conditions of entity integrity and referential integrity and implement them in a database. E.g. students independently create primary and foreign keys on a computer, as well as corresponding constraints. (I6)
- In the practical preliminary exam and/or quiz, students independently modify the existing database or create simple and complex queries in a database using a chosen query language. E.g. students independently add a new attribute to the existing relation, create a view of the existing relation or find all the information on students whose average grade in a given academic year was over 4.3. (I7)
- On a computer, students independently and/or according to the instructions create a software solution with user interface using a chosen tool, on the basis of a previously created database. E.g. on a computer, students independently create a complex report from the relational database or a user interface for working with the existing relational database, all using a chosen programming tool. (I8)

10. Mandatory literature (at the time of submission of study programme proposal)

1. C. J. Date (2012). Database Design and Relational Theory: Normal Forms and All That Jazz. O'Reilly Media.
2. C. J. Date (2015). SQL and Relational Theory: How to Write Accurate SQL Code. O'Reilly Media.
3. Poščić, P. (2018). Databases, scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. M. Varga (1994). Baze podataka; konceptualno, logičko i fizičko modeliranje podataka. DRIP, Zagreb.
2. M. Radovan (1993). Baza podataka - relacijski pristup i SQL. Informator, Zagreb.
3. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Sanja Čandrlić, PhD	
Name of the course	Introduction to Software Engineering	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to basic concepts, methods, techniques and principles from the field of software engineering and to develop an engineering approach and encourage teamwork in software development projects.

2. Course enrolment requirements

A pass mark in Programming 1 and previously taken Data Modelling.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Create developer and user documentation and perform configuration management of such documentation.

I2. Compare life cycle and development process models and choose an appropriate method for an engineering approach to software development.

I3. Explain the relationship between non-functional and functional user requirements on a specific example and propose methods to solve them.

I4. Determine basic elements of a user interface based on user requirements.

I5. Plan the development of components, design components and plan their integration into the system.

I6. Describe basic concepts of software testing.

I7. Plan and create an application prototype using a given development environment and manage configurations.

4. Course content

- Concept of software engineering. Historical overview. Formal principles of software engineering. Methods and stages of software system development.
- Requirements analysis and specification. Non-functional and functional user requirements. System modelling. Designing a user interface.
- Designing a system architecture. Designing program modules. Programming objectives and techniques. Prototyping and fast application development. Use of CASE tools. Verification and validation.
- Evolution and software system maintenance. Software reuse.

- Configuration management. Software re-engineering. Quality assurance. Documenting of software systems.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment and successfully pass it.
- Undertake practical problem-solving tasks and successfully complete them.
- Make a project, independently or in groups, and present it to the lecturer.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[24] *of student work*

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	2	Continuous assessment		Report		Practical work	0.5
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts of engineering (I2, I3, I6). E.g. through multiple choice questions, fill in the blank questions and essay questions, they list the characteristics, advantages and disadvantages of different development life-cycle models and reference development processes, as well as their selection criteria for different problem classes.
- A project in which students, individually or in groups, need to develop new system models, plan the development of system components, sketch and design a user interface and create an application prototype (I4, I5, I7). Students present their solution to the lecturer.
- As part of the project or practical problem-solving tasks (subprojects) related to different stages of application development, students need to prepare documentation regarding project management,

[24] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

program architecture, code, testing, user instructions and support and maintenance instructions, applying the appropriate standards and developer tools (I1, I5, I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Van Vliet, H.: Software Engineering - Principles and Practice, 3rd Edition. John Wiley&Sons, Chicester UK, 2008.
2. Manger, R. Softversko inženjerstvo, Element, Zagreb, 2016.
3. Bourque, P., Fairlez, R. E. SWEBOK v 3.0 – Guide to the Software Engineering Body of Knowledge, IEEE, 2014.
4. Group of authors. Joint Course on Software Engineering, Online script with lectures in Moodle e-course, 2016.
5. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Sommerville, I.: Software Engineering, 10th Edition, Pearson Education, London, 2016.
2. McConnell, S. Code Complete: A Practical Handbook of Software Construction, MicrosoftPress, 2004.
3. Pressman, R. Software Engineering: A practitioner's Approach, McGraw-Hill, New York, 2014.
4. Jones, C. Software Engineering Best Practices, McGraw-Hill, 2010.
5. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Martina Holenko Dlab, PhD	
Name of the course	Operations Research	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about problem formalization processes related to optimization and allocation, as well as about methods for determining and analysing their solutions for the purpose of making decisions in the business environment.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe basic concepts of operations research and problem-solving methods related to optimization and allocation.

I2. Identify the type and characteristics of linear problems related to optimization and allocation.

I3. Set up a mathematical model for a given linear problem related to optimization and allocation.

I4. Solve linear programming problems with graphical and simplex methods, using software.

I5. Solve linear problems related to allocation using appropriate methods and software.

I6. Apply the concepts of linear dependence and linear independence of vectors and methods of linear algebra to solve problems in the field of information sciences using software.

I7. Analyse linear problems in the field of information sciences and their solutions to support the process of making business decisions.

4. Course content

- Concept and development of operations research. Process of solving operations research problems.
- Linear programming. Setting up a mathematical model for linear programming problems.
- Solving linear programming problems with graphical method.
- Solving linear programming problems with simplex method.
- Degeneracy.
- Duality. Dual simplex method. Sensitivity analysis.
- Transportation problem. Methods for setting up the initial solution of a transportation problem. Methods for testing the initial solution and finding an optimal solution of a transportation problem.
- Allocation problem. Methods for solving allocation problems.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Actively participate in practical problem solving during auditory and laboratory exercises.
- Complete problem-solving tasks as part of homework and hand in the solutions within the deadline. Preliminary exams are preceded by homework. After doing their homework, students receive feedback that can help them prepare for the preliminary exam.
- Participate in continuous assessment (preliminary exams) and successfully pass them.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[25] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	1
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- I1, I2 – as part of their homework, preliminary exams and the final exam, taking place in the form of online assessment, students are required to demonstrate their knowledge of basic concepts related to operations research and problem-solving process and to identify a given problem, its properties and propose methods for their solution, all by answering questions (e.g. multiple choice questions, fill in the blank questions, essay questions).
- I3, I4, I5, I6 – as part of their homework, preliminary exams and the final exam, taking place in the form of online assessment, students are required to hand in the files with solutions to problem-solving tasks, thus showing their ability to set up a mathematical model and find a solution to a given problem

[25] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

using software.

- I7 – as part of their homework, preliminary exams and the final exam, taking place in the form of online assessment, students are required to interpret and analyse the obtained results of problem-solving tasks (practical problems) and prepare quantitative grounds for decision-making using software, all by answering questions (e.g. multiple choice questions, fill in the blank questions, essay questions).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Hillier, F. S., Lieberman, G. J. Introduction to operations research. Tata McGraw-Hill Education, 2012.
2. Winston, W. L., Goldberg, J. B. Operations research: applications and algorithms. Belmont: Thomson Brooks/Cole, 2004.
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Murthy, G. S. R. Applications of Operations Research and Management Science, Springer, 2015.
2. Z. Lukač, L. Neralić, Operacijska istraživanja, Element, 2012.
3. D. Barković, Operacijska istraživanja, Sveučilište J. J. Strossmayera u Osijeku, Ekonomski fakultet, Osijek, 2001.
4. D. Kalpić, V. Mornar, Operacijska istraživanja, Zeus, Zagreb, 1996.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Mario Radovan, PhD / Igor Jugo, PhD	
Name of the course	Introduction to Web Programming	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about technologies, planning and creation of web application architecture, methods for creating basic templates for user interaction with a web application for the purpose of performing basic functionalities: create, read, update and delete – CRUD in a selected data warehouse. Students will be taught how to independently analyse requests, plan a basic web application architecture, a data warehouse (relational database or text files (text, XML, JSON) and create an interactive web application using client-side and server-side scripting languages, and how to carry out a performance analysis.

2. Course enrolment requirements

Previously taken course Programming 1.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse properties of programming languages and frameworks (platforms) for developing back-end web applications.

I2. Analyse properties of programming languages and frameworks (platforms) for developing front-end web applications.

I3. List and describe key principles and methods of server-side programming, which is the basis for web application operation.

I4. Debug web application code.

I5. Design web application architecture with the choice of appropriate technologies (web server, programming languages and frameworks (platforms), data warehouse etc.) based on a given description (list of requests).

I6. Build an interactive web application using the chosen technologies.

I7. Test the website workload using specialised tools on the basis of developed plan.

I8. Propose improvements to the web application based on website workload analysis and described latency using a timing diagram.

4. Course content

Basic concepts – essential technologies for the operation of WWW and development of interactive web applications, main challenges, introduction to web engineering.

Syntax of scripting language – operators, branching, iterations, functions, working with arrays, strings, files, date and time.

Basic interaction templates and dynamic generation of web application elements.

Data warehouses for web applications – files and databases.

Basic web application operations (CRUD).

Basics of client-side scripting for the purpose of increasing application interactivity and security.

Web application performance analysis, overview of possibilities to improve performance, structural and non-structural changes.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, computer laboratory work, individual work outside the computer laboratory and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams).
- Design, create and present a solution to a problem-solving task.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[26] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts. (I1-I8)

[26] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- Practical assessment (practical preliminary exam) in which students implement a given web application on a computer using a given programming language. (I3-I6)
- Practical assessment (practical preliminary exam) in which students implement a workload plan and suggest improvements to a given web application using a given programming language. (I7-I8)
- Practical project assignment in which students apply the acquired knowledge and skills to complete problem-solving tasks such as development of simple web applications or web application modules related to the topic of their choice, according to the instructions and evaluation criteria set in advance. (I1-I8)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Welling, L., Thompson, L.: PHP and MySQL Web Development 5th Edition (2016), Sams Publishing.
2. Nixon, R.: Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, 5th Ed (2018), O'Reilly Media.
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Marija Brkić Bakarić, PhD	
Name of the course	Algorithms and Data Structures	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about algorithm design strategies by solving given problems and to introduce them to abstract data types such as tree and graph, along with the analysis of time and space complexity.

2. Course enrolment requirements

A pass grade in Programming 1 and Programming 2.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students will be able to:

I1. Apply the accounting method, aggregate analysis method and potential method to determine amortized complexity of a data structure.

I2. Compare a priori and a posteriori analyses of time complexity.

I3. Determine time and space complexity of algorithms by estimating the growth function.

I4. Illustrate basic algorithms on given linear lists, trees and graphs.

I5. Implement the solution to a given problem by applying an appropriate algorithm design strategy (divide and conquer, dynamic programming, greedy algorithms, backtracking).

I6. Implement own and use available linear (linked list, stack, queue) and/or tree (binary tree, binary search tree, heap, general tree) abstract data types to solve a given problem.

I7. Recognise and solve a problem by implementing an appropriate algorithm on the graph abstract data type.

4. Course content

Principles of algorithm analysis. Algorithm design strategies (divide and conquer, dynamic programming, greedy algorithms, backtracking). Tree. Binary search tree (AVL tree, red-black tree, k-d tree). Heap. Multi-branch tree (B-tree). Graph properties and types. Path on a graph. Algorithms on graphs (connectivity algorithm, minimal spanning tree, shortest path algorithm).

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________

6. Comments	E-learning system will be used in this course. When they enrol into the course, students will be instructed to use the tools available in the system. A detailed schedule will be provided in the syllabus.

7. Student responsibilities

Students should regularly participate in all course activities (preliminary exams, projects, problem-solving tasks, homework) and score at least 50% on the final exam. A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[27] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam	0.5	Essay		Research	
Project	0.5	Continuous assessment	1.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts related to algorithm analysis, algorithm design strategies and abstract data types such as trees and graphs (I1, I2, I4), e.g. By applying the Dijkstra's algorithm determine the shortest path tree for the shown graph if node A is the root. Choose an answer that shows the sequence of adding nodes to the tree.

Written or online assessment (theoretical preliminary exam) in which students analyse different code segments and determine time and space complexity (I3), e.g. determine and choose the complexity of a given code segment.

Practical assessment (practical preliminary exam) in which students need to design an algorithm for solving a given problem and implement it using a given programming language (I5), e.g. You go to a supermarket and you have a backpack of volume n at your disposal. You have to fill it with as much food as possible to be able to survive in an atomic bomb shelter. Information on products at your disposal is written in the file *toy.txt* in format <volume value>.

Practical assessment (practical preliminary exam) in which students need to choose an appropriate abstract data type to solve a given problem and implement the solution using a given programming language (I6, I7), e.g. Write a program that simulates the operation of emergency services, enabling the entry of a new patient, processing of the one next in line and printout of patient data.

Group or individual practical project assignment in which students apply knowledge and skills from algorithm and data structure theory to solve a practical problem-solving task according to the set instructions and evaluation criteria (I5, I6, I7), e.g. Compile a frequency dictionary based on a given corpus.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Scripts, presentations and other learning material available in the e-course.
2. Richard F. Gilberg, Behrouz A. Forouzan: Data Structures: A Psuedocode approach with C, Cengage Learning, 2004.
3. Robert Sedgewick, Kevin Wayne: Algorithms, Parts 1-2, Addison-Wesley Professional, 2014.
4. Michael T. Goodrich, Roberto Tamassia: Algorithm Design and Applications, John Wiley & Sons, Inc., 2015.

[27] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms, 2nd edition, The MIT Press, 2001.
2. Varsha H. Patil: Data Structures Using C++, Oxford University Press, 2012.
3. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani: Algorithms, McGraw-Hill, 2008.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Sanda Martinčić-Ipšić, PhD	
Name of the course	Information Technology Project Management	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about ICT project management, project planning, making project studies and teamwork.

2. Course enrolment requirements

Previously taken course Introduction to Software Engineering.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Apply project management techniques and methods for projects consisting of development and introduction of information and communications technology (ICT).

I2. Define and quantify project goals and sub-goals and associate activities, project deliveries and responsibilities with them.

I3. Prepare project documentation including a detailed execution plan (time schedule, budget, necessary human resources and communication plan).

I4. Identify risks in the implementation of ICT projects and make a risk management plan draft.

I5. Elaborate and implement the project delivery implementation, control and acceptance plan.

I6. Evaluate performance factors of ICT projects with respect to the application of specific technology and ICT system development methods in business environment.

4. Course content

- Project definition. Goals, deadlines, resources and constraints. Project organization and mode of operation. Types of projects.
- Specificities of ICT projects. Methods of ICT project management.
- Project phases. Project activity planning. Network planning. PERT, CPM, Gantt charts. Critical path analysis. Cost analysis. Resource analysis. Supervising the implementation of a project. Delivery plan.
- Risk management. Change management. Quality assurance. Standardization and documentation.
- Project teams. Types of teams. Difference between team and work group. Team roles. Roles in information technology development projects. Communication plan and team organization.
- Project manager roles and responsibilities. Motivation, communication and conflict resolution. Techniques for fostering creativity in a team.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☒ mentorship
	☒ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system. Fieldwork is organised according to the possibilities (e.g. visiting companies or inviting IT project management experts, studying cases and examples from practice).	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow and take part in course activities.
- Participate in continuous assessment (preliminary exams).
- Do homework and complete individual assignments during exercises.
- Create (individually or in teams) a project study including all the elements listed under learning outcomes, present it and write comments on other students' project studies.
- Written (or online) final exam for the course covers all the course material and requires a score of 50% to pass.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[28] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1.5	Continuous assessment	1	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

Learning outcomes are assessed in writing and (or) orally, and especially through practical work on specific project assignments in class or during practical training, as follows:

- Students need to individually prepare a project assignment for a specific example of an ICT project. (I1)
- On a specific example of an ICT project, students need to apply project management techniques and methods (e.g. SWOT, activity/problem/goal decomposition diagram, create a feasibility study,

[28] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Gantt chart, financial plan, delivery plan etc.). (I1, I2, I3, I4, I5, I6)

- Students define and quantify project goals for a specific problem and associate activities, project deliveries and responsibilities with them. (I2)
- Students prepare project documentation for a specific project, which includes a detailed execution plan with execution dynamics, necessary budget and human resources, as well as communication plan within the project team and with users and sponsors. (I3)
- For a given ICT project, students identify risks in the implementation of ICT projects and make a risk management plan draft. (I4)
- Students need to elaborate project delivery implementation, control and acceptance plan for a given ICT project. (I5)
- For a given ICT project, students need to evaluate performance factors of ICT projects with respect to the application of certain technology and development methods. (I6)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Learning material available in the e-learning system, together with own notes and materials from lectures and exercises.
2. Krešimir Fertalj, Željka Car, Ivana Nižetić Kosović, Upravljanje projektima, FER, Zagreb, 2016. https://bib.irb.hr/datoteka/807419.Upravljanje_projektima_-_skripta_FER_2016.pdf
3. Robert Wysocki, Effective Project Management: Traditional, Agile, Extreme. 7th edition, John Wiley & Sons, 2014.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. A Guide to the Project Management Body of Knowledge, PMI, 6th edition, 2017.
2. Harold Kerzner: Project Management: A System Approach to Planning, Scheduling and Controlling, John Wiley & Sons, New Jersey, 2017.
3. Kathy Schwalbe, Information Technology Project Management, Revised 7th Edition, Cenage, 2014.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Mile Pavlić, PhD	
Name of the course	Information Systems Development	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to teach students how to create project documentation for information system development and necessary software solution components. For a selected new development environment, a transition of business processes, applications and documentation is planned, as well as database migration to a new system.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Choose and apply an information system design method appropriate for a given problem, which includes approaches, processes, methods and techniques.

I2. Embed the requirements into a model for a new or improved system for a given business area.

I3. Formulate support, automation and improvement possibilities based on the application of ICT in a certain business area (production, logistics, healthcare, financial institutions etc.).

I4. Plan the development and implementation of an information system, as well as application design, including risk analysis and performance indicators, by using project management methodology.

I5. Build a future business model and information system model and explain them to stakeholders by using process and data modelling and organizational design.

I6. Choose a development environment and ICT necessary to build a software solution and information system or system components in line with the project and financial and technical resources.

I7. Plan information system migration and user training.

4. Course content

- Methodologies, methods, models and tools for designing information systems.
- User requirements analysis, modelling of processes and data contained in requirements, expanding of the existing models with new requirements.
- Planning information system and application development, planning subsystems and connections, determining priorities, choosing an ICT, risk management.

- Designing software product architecture, planning software production activities.
- Designing new databases, planning database migration.
- Software production activities. Testing.
- Implementation, application and maintenance.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ___________________
6. Comments	Classes are held by combining classroom work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment and achieve the number of credits equal to or higher than the passing score (if any).
- Participate in practical problem-solving tasks and achieve the number of credits equal to or higher than the passing score (if any).
- Individually or in teams, make a project and present it to the lecturer, and achieve the number of credits equal to or higher than the passing score (if any).

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[29] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1.5	Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	
Portfolio		Application					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts from the field of information systems development (I1, I2, I3, I4), e.g. through multiple choice questions, fill in the blank questions and essay questions they list the characteristics of different methodologies for information systems development.

[29] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- A project in which students, individually or in groups, need to choose methods for the collection of user requirements, create process and data (ERD and DFD) models and software product architecture. Students present their solution to the lecturer (I4, I5). For example, after they have collected user requests using a method of their choice, students create ERD, DFD and software product architecture and present their solutions to the lecturer and other students.
- Practical problem-solving tasks in which students define a development plan and migration (I6, I7). For example, define a development plan and migration from an "outdated" ICT level to a new level.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Scripts, presentations and other learning material available in the e-course.
2. Pavlić, M., Informacijski sustavi, Školska knjiga, Zagreb, 2011.
3. Valacich J. S., George J. F Modern Systems Analysis and Design. 8th ed. Pearson Education, Inc, 2017.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Pavlić, M., Oblikovanje baza podataka, Odjel za informatiku, Sveučilište u Rijeci, Rijeka, 2011.
2. Pavlić, M., Jakupović, A., Čandrlić, S. Modeliranje procesa, Odjel za informatiku, Sveučilište u Rijeci, Rijeka, 2014.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Božidar Kovačić, PhD	
Name of the course	Security of Information and Communication Systems	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge from the field of information system security, to introduce them to risks and threats to information systems, methods for their protection, data encryption and decryption methods, as well as procedures for measuring and evaluating the achieved level of information security.

2. Course enrolment requirements

Previously taken courses Fundamentals of Informatics and Computer Networks.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse protocols in secure and insecure communication channels.
I2. Define and explain differences between HTTP and HTTPS protocols.
I3. Determine protection functionalities of an information system, and build an information system with authentication, authorization and journal modules.
I4. Assess information security risks for personal computers and servers and describe the methods of potential attacks.
I5. Explain the methods of protecting information systems from certain types of attacks on data integrity.

4. Course content

- Information system security risks. Risk analysis and assessment. Security threats and their probability. IS vulnerability.
- Security incidents in an information system. Detecting signs of security incidents.
- Security mechanisms and control procedures, cryptography, data encryption and decryption.
- Information security system management, improvement and monitoring. Measuring control efficiency.
- Security risk management. Risk assessment methods. Risk management as an instrument for improving security.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories

	□ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). A detailed schedule with lectures and exercises will be defined in the syllabus. When they enrol into this course, students will be instructed to use the distance learning system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Write an individual or group paper on a given topic and present it to lecturers and other students.
- Score at least 50% on the final exam.
- A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[30] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1.5	Continuous assessment	1	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- In the oral or written exam, students explain the properties of and differences between protocols in secure and insecure channels using a computer, and provide valid reasons for implementing a certain protocol (I1).
- In the written exam, students explain the differences between HTTP and HTTPS protocols in a well-argued manner (I2).
- In the practical project assignment, students determine protection functionalities for a specific information system, and create or configure user authentication and authorization model for a given application (I3).
- As part of the project, students need to work out the development of a new IT service, and perform a risk analysis (I4).
- As part of the project, students need to work out the development of a new IT service, and manage the service level, incidents, problems, requests and availability (e.g. make a response table by problem and incident type) (I3, I5).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Dieter Gollman, "Computer Security", John Wiley & Sons, 2011.

[30] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

2. Harold F. Tipton, Micki Krause, "Information Security Management", 6th edition, Taylor & Francis Group, 2007.
3. Information Security Policies and Procedures: A Practitioner's Reference, Second Edition, Thomas R. Peltier, 2004.
4. Wenliang Du (Author), "Computer Security: A Hands-on Approach", Create Space, 2017

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Donald L. Pipkin, "Information Security", Prentice Hall PTR, 2000
2. Thomas R. Peltier, "Information Security Risk Analysis", Third Edition, CRC Press, 2010.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Marina Ivašić-Kos, PhD	
Name of the course	Programming Paradigms and Languages	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for SD module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The course provides the overview of different programming paradigms. It tackles concepts existing in various imperative and declarative programming languages. Object-oriented programming paradigm for mobile devices is addressed in more detail and students are introduced to the visual paradigm.

2. Course enrolment requirements

A pass mark in Programming 1, Programming 2 and Object-Oriented Programming.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Distinguish between different programming paradigms (imperative, declarative, object-oriented, visual) and describe the main differences between them.

I2. Compare and analyse different programming languages and classify them by paradigms to which they belong.

I3. Identify equivalent concepts in different programming languages and paradigms.

I4. Choose an appropriate programming paradigm to solve a specific problem-solving task.

I5. Apply basic knowledge from the field of object-oriented paradigm and theoretical bases related to design and development of mobile applications in analysing and comparing mobile applications.

I6. Independently work out and create an application using appropriate concepts from the field of object-oriented paradigm.

4. Course content

- Basic programming paradigms (imperative, declarative, object-oriented, visual) and programming languages. Other paradigms (visual, parallel programming, component-oriented programming, generic, scripting). Criteria for the classification of programming languages. Classification of programming languages.
- **Imperative paradigm**: development, basic properties, programming language C.
- **Functional paradigm**: theoretical basis, basics of lambda calculus, evaluation strategies, pattern matching, programming language Haskell.
- **Logic paradigm**: theoretical basis – first-order logic, deduction, unification theory, programming language Prolog.

- **Object-oriented paradigm**: development, theoretical basis, programming language Java.
- **Programming for mobile devices**: Introduction to the development of Android applications. Development environment, basic architecture and life-cycle. System components (Intents, Activities, Services, Content Providers, Broadcast Receivers). Basic elements of user interface (navigational components, input controls, menus, action bar, fragments, dialogs, notifications). Event handling. View controllers and components. XML language and working with resources. Advanced concepts: location services, maps and use of sensors. Saving data.
- **Component-oriented, parallel and distributed programming** – theoretical basis and practical approach through Java.
- **Scripting and constraint programming** – theoretical basis and practical approach through Python.
- **Generic programming** – theoretical basis and practical approach through C++ .
- **Visual programming** – theoretical basis and practical examples.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________

6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams).
- Design, create and present a solution to a problem-solving task, individually or in pairs, and score at least 50% on the final exam.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[31] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1.5

[31] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts related to different programming paradigms, analyse and classify programming languages into appropriate paradigms and compare equivalent concepts in different programming languages (I1-I3).
- Practical assessment (practical preliminary exam) in which students choose an appropriate programming language and work out an algorithm for completing a specific problem-solving task on a computer (I3, I5).
- Final exam: Practical project assignment in which students apply their skills and knowledge from the field of specific programming paradigm to solve problem-solving tasks related to the topic of their choice (individually or in pairs), according to instructions and evaluation criteria set in advance (I7).

10. Mandatory literature (at the time of submission of study programme proposal)

1. M. Gabrielli, S. Martini: Programming Languages: Principles and Paradigms, Springer, 2010
2. B. A. Tate: Seven Languages in Seven Weeks, Pragmatic Bookshelf, 2010
3. Ed. Burnette: Hello, Android: introducing Google’s mobile development Platform (2nd edition),2015.
4. B. Phillips, C. Stewart: Android Programming: The Big Nerd Ranch Guide (4th Edition), 2019
5. M. Ivašić-Kos: Razvoj android aplikacija: on-line prezentacije predavanja, zadaci i primjeri riješenih zadataka, Moodle e-knjiga, 2019
6. M. Ivašić-Kos: Objektni programski jezici – Java, on-line prezentacije predavanja, zadaci i primjeri riješenih zadataka, Moodle e-knjiga, 2018

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. A. B. Tucker, R. E. Noonan: Programming Languages – Principles and Paradigms (2nd ed.), McGraw-Hill, 2007
2. D. P. Friedman, M. Wand, C. T. Haynes: Essentials of Programming Languages, 2/e, MIT Press, 2001
3. S. McConnell: Code Complete: A Practical Handbook of Software Construction, 2/e, MS Press, 2004
4. B. Stroustrup: Programming -- Principles and Practice Using C++ (Second Edition), Addison-Wesley, 2014
5. T. Petricek, J. Skeet, Real World Functional Programming: With Examples in F# and C#, Manning, 2010

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Miran Pobar, PhD / Vedran Miletić, PhD	
Name of the course	Code Optimization	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for SD module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of this course is to introduce students to basic principles and methods of code optimization at the level of abstract syntax, program flowchart and executable (machine) code.

2. Course enrolment requirements

A pass mark in Algorithms and Data Structures.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse the properties enabling code transformation and represent the code using a flowchart.

I2. Show the differences between local and global optimization and identify where each of them applies.

I3. Perform a conventional data flow analysis, register allocation by graph colouring and common subexpression elimination.

I4. Describe mode of operation of higher-level optimization and apply existing optimizations.

I5. Describe differences between higher-level optimizations and target architecture-specific optimizations.

I6. Choose instructions.

I7. Analyse the problem of optimization phase sequence.

4. Course content

- Overview of programming language optimizing compiler. Optimization per elements. Analysis of properties enabling transformation. Flowchart and representation of program concepts. Problem of optimization phase sequence.
- Types of optimization. Local optimization: peephole optimization, instruction scheduling. Global optimization: common subexpressions, code changes. Interprocedural optimization. Call graph.
- Conventional data flow analysis. Algorithms on graphs, sets of live and available variables. Register allocation by graph colouring. Common subexpression elimination. Spilling to memory; use of temporary expressions introduced during common subexpression elimination. Data flow anomalies. Static single assignment form.
- Overview of higher-level optimizations. Pointer analysis and pseudonym analysis.
- Target architecture-specific optimization. Choice of instruction. Instruction scheduling and related problem of optimization phase sequence.

5. Manner of instruction	☒ lectures	☒ individual assignments

	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Complete individual or team practical work related to a given topic.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[32] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	1
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Practical assessment on a computer (practical preliminary exam), in which students analyse and transform the code, and use and adapt the existing optimizations (I1, I2, I3, I4, I6).
- Group or individual practical work in which students, according to the set instructions, implement a solution containing required optimizations and draw up documentation for their own implementation (I1, I2, I3, I4, I6).
- Written or online assessment in which students demonstrate their understanding of theoretical concepts related to optimization of programming code, e.g. through multiple choice questions, fill in the blank questions and essay questions (I1, I2, I4, I5, I7).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Cooper, K. D. & Torczon, L. Engineering a compiler. (Elsevier/Morgan Kaufmann, 2011).
2. Holub, A. I. Compiler design in C. (Prentice Hall, 1990). (E-book is available for free download from the author’s site http://holub.com/compiler/ and can be printed if necessary)

[32] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Fraser, C. W. & Hanson, D. R. A retargetable C compiler: design and implementation. (Benjamin-Cummings, 1995).
2. Muchnick, S. S. Advanced compiler design and implementation. (Morgan Kaufmann, 1997).
3. Nielson, F., Nielson, H. R. & Hankin, C. Principles of program analysis. (Springer, 1999).
4. Appel, A. W. Modern compiler implementation in C. (Cambridge University Press, 2004).
5. Aho, A. V., Lam, M. S., Sethi, R. & Ullman, J. D. Compilers: principles, techniques, & tools. (Pearson/Addison-Wesley, 2006).
6. Morgensen, T. Ae. Basics of Compiler Design. (Lulu, 2010).
7. Wilhelm, R. & Seidl, H. Compiler design: virtual machines. (Springer, 2011).
8. Hack, S., Wilhelm, R. & Seidl, H. Compiler design: code generation and machine-level optimization. (Springer, 2019).
9. The GNU Compiler Collection. GCC online documenatation. (GNU, 2019). (available online: https://gcc.gnu.org/onlinedocs/)
10. The LLVM Compiler Infrastructure. LLVM documentation. (LLVM, 2019). (available online: https://llvm.org/docs/)

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Božidar Kovačić, PhD / Igor Jugo, PhD	
Name of the course	Web Programming	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for SD module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master knowledge about advanced concepts related to interactive web application development (advantages and disadvantages of individual development approaches – procedural, OO, MVC, MVMM, DESIGN PATTERNS), use of JavaScript technologies for web application development, and to acquire technologies and skills enabling web programmers to work independently – security concepts in web programming, code testing and documenting, team work in version control systems.

2. Course enrolment requirements

A pass mark in Introduction to Web Programming.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Create a database by using MySQL relational system and be familiar with possibilities for its use in dynamic web applications.

I2. Identify and analyse different approaches to web application development.

I3. Create a web application according to the given functional and non-functional requirements by using different approaches (procedural, OO, MVC, MEAN).

I4. Identify advantages and disadvantages of individual approaches with respect to software solution quality.

I5. Recognize individual design patterns and identify advantages of implementation of certain patterns.

I6. Develop a set of tests and run these tests to control the software solution quality.

I7. Generate documentation for the software solution.

I8. Apply teamwork techniques in version control systems.

4. Course content

- Object-oriented modelling of web applications and object-oriented web programming in scripting languages: classes, objects, methods; constructors, inheritance, overriding.
- Relational data model and relational databases: defining structural properties of tables and databases; creation of tables and databases. SQL language, creation of combined SQL expressions (commands).
- Two-way communication between client, server (dynamic web application), interpreter and database system.

- Creating a communication interface for access to a dynamic web application; creating a connection between scripts and database systems, accepting and forming replies from a database.
- Queries and changing database content using a scripting language.
- Scripting with JavaScript technologies.
- Security of dynamic web applications.
- Writing and implementing automated code tests.
- Writing and generating code documentation.
- Design patterns in web applications.
- Teamwork in version control systems.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, computer laboratory work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams).
- Design, create and present a solution to a problem-solving task.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[33] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

[33] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- Written or online assessment (theoretical preliminary exams) in which students demonstrate their understanding of theoretical concepts. (I1-I8)
- Practical assessment (practical preliminary exams) in which students implement a given web application on a computer using a given programming language. (I1-I7)
- Practical project assignment in which students, in teams, apply the acquired knowledge and skills to solve problem-solving tasks, for example, develop a complete web application related to the topic chosen by the team, according to the instructions and evaluation criteria set in advance. (I1-I8)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Scripts, presentations and other learning material available in the e-course.
2. Lockhart, J., **Modern PHP: New Features and Good Practices**, 2015, O'Reilly Media
3. Zandstra, M., **PHP Objects Patterns and Practice**, APress, 5th edition, 2016.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. https://manual.phpdoc.org
2. Bergman, S., **PHP Unit**, O'Reilly, 2005,
3. Chacon, S., Straub, B., **Pro Git**, APress, 2nd edition, 2014

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Mario Radovan, PhD / Vedran Miletić, PhD	
Name of the course	Communication Networks	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for CS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about devices, protocols and standards in the domain of communication networks and their application in creating different types of networks, with a focus on networks in data centres.

2. Course enrolment requirements

A pass mark in Computer Networks.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Classify and compare computer networks.

I2. Explain how a network switch manages congestion, flow and traffic.

I3. Analyse computer networks in data centres by using device documentation and protocols, as well as software tools.

I4. Name advantages and disadvantages of individual network topologies for the purpose of choosing an optimal topology when building or upgrading a communication network.

I5. Arrange network virtualization and network function virtualization by using appropriate software tools.

I6. Recognise and express development trends in the information and communications technology in the domain of data centre communication networks.

4. Course content

- Evolution of communication networks. Devices, network protocols and standards.
- Switch architecture. Switch topology. Congestion management. Flow management. Traffic management.
- Data centre network topologies. Data centre network standards.
- Network virtualization. Separating network management functions and network data functions. Network function virtualization.
- Development trends in data centre communication networks.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship

	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Complete problem-solving tasks as part of homework and hand in the solutions within the deadline.
- Answer questions and complete problem-solving tasks during auditory and/or laboratory exercises.

  Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[34] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	1
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- As part of their homework, in the form of online assessment, students are required to hand in their files with solutions to problem-solving tasks, demonstrating their ability to set up a network model and/or required network service configuration by using a network simulator, network emulator and server and client applications (I3, I5).
- In laboratory exercises, students are required to take written or online assessment, in which they demonstrate their understanding of theoretical concepts necessary to complete programming tasks, e.g. through multiple choice questions, fill in the blank questions and essay questions (I2, I4). Following the written or online assessment, students are required to solve problem-solving tasks and hand in the files with solutions through online assessment, thus demonstrating their ability to set up a network model and/or required network service configuration by using a network simulator, network emulator and server and client applications (I3, I5).
- Written or online assessment in which students demonstrate their understanding of theoretical concepts related to communication networks, e.g. through multiple choice questions, fill in the blank questions and essay questions (I1, I2, I4, I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Kurose, J. F. & Ross, K. W. Computer networking: a top-down approach. (Pearson, 2013).
2. Peterson, L. L. & Davie, B. S. Computer networks: a systems approach. (Morgan Kaufmann, 2012).
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

[34] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

1. Bažant, A., Gledec, G., Ilić, Ž., Ježić, G., Kos, M., Kunštić, M., Lovrek, I., Matijašević, M., Mikac, B. & Sinković, V. Osnovne arhitekture mreža. (Element, 2014).
2. Halsall, F. Computer networking and the Internet. (Addison-Wesley, 2006).
3. Tanenbaum, A. S. & Wetherall, D. Computer networks. (Pearson/Prentice Hall, 2011).
4. Sterbenz, J. P. G. & Touch, J. D. High speed networking: a systematic approach to high-bandwidth low-latency communication. (Wiley, 2001).
5. Comer, D. Computer networks and Internets. (Pearson, 2015).
6. Comer, D. Internetworking with TCP/IP. (Pearson/Prentice Hall, 2013).
7. Lee, G. Cloud Networking: Understanding Cloud-based Data Center Networks. (Morgan Kaufmann, 2014).

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Božidar Kovačić, PhD	
Name of the course	Network and Mobile Operating Systems	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for CS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about network operating systems and mobile operating systems, as well as knowledge about services in network operating systems: process execution, hardware detection, protocol execution, web services and security, and knowledge about basic concepts of mobile operating systems: activity management, communication, services and security.

2. Course enrolment requirements

A pass mark in Operating Systems.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Name and explain main tasks of network operating systems with respect to the structure of network operating systems.

I2. Analyse network operating system services and choose services for individual functions of network operating systems according to the set specifications.

I3. Classify and compare protocols for the purpose of web services in network operating systems.

I4. Analyse security mechanisms for network operating systems and justify the use of basic and additional protection functionalities for a specific network operating system.

I5. Name and explain basic tasks of mobile operating systems with respect to the structure of mobile operating systems.

I6. Explain methods of activity management when running applications on mobile operating systems.

I7. Link parts of mobile operating system and hardware used for managing input-output units.

I8. Analyse security mechanisms for mobile operating systems and justify the use of basic and additional protection functionalities for a specific operating system.

4. Course content

- Functions of network operating systems: support to process execution, hardware detection, protocol execution, web services, security.
- Implementation of file system, naming and replicas in network operating systems.
- Network protocols and supported web services in network operating systems.
- Security of network operating systems: authentication and authorisation, resource access control, system restrictions.

- Mobile operating system architecture.
- Activity management when running applications on mobile operating systems.
- Communication in mobile operating systems: processing user requests using a touch screen, managing sensors used in mobile devices, supported web services.
- Mobile operating system services.
- Security of mobile operating systems.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories
	☐ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures and laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Prepare an individual or group project on a given topic, in a written form, and present it to lecturers and other students.
- Score at least 50% on the final exam.
- A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[35] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam		Oral exam		Essay		Research	
Project	0.5	Continuous assessment	2	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

[35] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- In the written exam, students should name types of network operating systems, sketch parts and connections of operating systems and provide explanations of basic tasks of operating systems (I1).
- In the practical assessment on a computer (completing a problem-solving task in the form of homework or during exercises to obtain credits), students complete a problem-solving task in which they need to choose and apply services necessary to implement a given functionality of the network operating system according to set specifications (I2).
- In the written or oral exam, students choose a method of application of a protocol for a specific web service and provide valid arguments for its use (I3).
- In the written exam, students identify security and protection mechanisms of a given network operating system (I4).
- In the written exam, students should name types of mobile operating systems, sketch parts and connections of operating systems and provide explanation of basic tasks of mobile operating systems (I5).
- In the practical project assignment, students identify the activity of mobile operating system and hardware during the processing with input-output units (I6- I7).
- As part of their homework, students work out a simple program for accessing and managing sensors used in a mobile device (I7).
- In the written exam, students identify security and protection mechanisms of a given operating system, and choose a specific mechanism for a given security problem (I8).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Philip Hunter, "Network Operating Systems", Addison-Wesley, 1995.
2. Arash Habibi Lashkari, "Mobile Operating Systems and Programming: Mobile Communications" VDM Verlag Dr. Müller, 2011.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Leo Budin, Marin Golub, Domagoj Jakobović, "Operacijski sustavi", Element, 2010.
2. Christian Benvenuti, "Understanding Linux Network Internals: Guided Tour to Networking on Linux", O'Reilly Media, 2009.
3. Reto Meier, Ian Lake "Professional Android", Wrox, 2018.
4. Jonathan Levin, "MacOS and iOS Internals", Technologeeks Press, 2016.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION

Course instructor	Assoc. Prof. Ana Meštrović, PhD	
Name of the course	Social Network Analysis	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for CS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about methods and procedures for social network analysis. The objective is to teach students how to apply social network analysis methods in specific cases.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain essential concepts from the field of social network analysis and procedures of social network analysis.

I2. Format data from a given data source into an appropriate graph or network form.

I3. Choose appropriate methods and techniques for analysing different types of social networks.

I4. Analyse and compare different models of social networks on a local, middle and global level.

I5. Apply appropriate methods and procedures of complex network analysis when analysing specific information from social networks (e.g. scientific collaboration networks).

I6. Create different types of visualization of social network data by applying tools and software for (social/complex) network analysis.

I7. Implement different models and procedures for predicting future connections in the network.

4. Course content

- Basic concepts from the field of social network analysis.
- Network models and model implementation.
- Network analysis at the global level: distance measures, clustering measures, degree distribution, network density, network connectedness, assortativity measures.
- Network analysis at the middle level: community identification (overview of community detection algorithms for a given network), motif analysis, graphlet analysis.
- Network analysis at the local level: centrality measures, identification of the most important network nodes, clustering measures.
- Network visualization by applying different models.

- Overview of algorithms for predicting future connections in the network.
- Ready-to-use tools and software for the analysis and visualization of social networks.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system, prepare for classes (do homework) and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical quizzes and preliminary exams) and successfully pass them.
- Complete individual or team practical work related to a given topic.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[36] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of social network analysis theory (I1), e.g. through multiple choice questions, fill in the blank questions and/or essay questions, students need to explain the concepts of small-world network, scale-free network, preferential attachment, how “hubs” occur etc.
- Practical assessment on a computer (completing a problem-solving task in the form of homework or

[36] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

during exercises to obtain credits) in which students need to choose an appropriate network/graph model and format data, construct a network for a specific problem (I2) and analyse a given network at the local, middle and global level (I3, I4, I5, I6), for example do the analysis for scientific collaboration network at the local, middle and global level.

- Practical assignment (practical preliminary exam) in which students apply procedures for predicting future connections in the network (by applying node similarity measures implemented e.g. in Python) (I7), e.g. evaluate algorithms for predicting future connections implemented in Python (Adamic/Adar, Jaccard Index, Preferential attachment etc.) on a given network.
- Practical project assignment in which students apply knowledge and skills from the field of social network analysis to complete problem-solving tasks according to the instructions and evaluation criteria set in advance (I3, I4, I5, I6, I7), e.g. to generate an appropriate network/graph form for the given Twitter data, analyse and visualise the network.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Meštrović, A. (2015). Online skripta: Analiza kompleksnih mreža, Odjel za informatiku, Sveučilište u Rijeci, Rijeka 2015.
2. Scott, J. (2017). Social network analysis. Sage.
3. Wasserman, S., & Faust, K. (1994). Social network analysis: Methods and applications (Vol. 8). Cambridge university press.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Newman, M. (2018). Networks. Oxford university press.
2. Russell, M. A. (2013). Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More. " O'Reilly Media, Inc.".
3. Carrington, P. J., Scott, J., & Wasserman, S. (Eds.). (2005). Models and methods in social network analysis (Vol. 28). Cambridge university press.
4. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Miran Pobar, PhD	
Name of the course	Multimedia Technologies	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for MMS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course if for students to acquire basic knowledge about multimedia data and technologies for their representation, processing, view and search.

2. Course enrolment requirements

A pass mark in Multimedia Systems.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Distinguish between different types and structures of digital multimedia content and multimedia file formats.

I2. Describe the model of human auditory and visual system.

I3. Describe specific hardware for working with digital multimedia content.

I4. Compare various media compression processes.

I5. Based on measurement results, compare image, audio and video compression processes and choose appropriate compression processes depending on the purpose.

I6. Compare multimedia search processes based on metadata and content.

4. Course content

Multimedia technologies and systems. Overview of media and data sources. Basics of coding and compression. Overview of current recording and compression standards. Principles of image compression, standards. Audio and video signal, occurrence and properties. Spatial, temporal and subjective redundancy. Audio and video signal compression and standards. Multimedia search based on metadata and content.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in	

	the system.

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Write an individual or group paper on a given topic and present it to lecturers and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[37] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	0.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts regarding multimedia systems (I1, I2, I3, I4, I6), e.g. through multiple choice questions, fill in the blank questions and essay questions students list the characteristics of standard multimedia formats, describe certain compression processes and name their characteristics.
- Practical assessment in which students need to apply various multimedia compression processes, measure their quality by using standard measures and choose an appropriate process on the basis of measured results (I4, I5).
- Group or individual seminar paper in the form of prepared project documentation for a chosen type of a larger project addressing in more detail a topic from the field of digital multimedia search or compression. Students will receive the instructions and evaluation criteria for the seminar in advance (I4-I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Li, Ze-Nian; Drew, Mark S.; Liu, Jiangchuan: Fundamentals of Multimedia, Second Edition, Springer, 2014.
2. T.M. Savage, K.E. Vogel, An Introduction to Digital Multimedia, 2008, Jones & Bartlett Publishers.
3. Blanken, H. M., de Vries, A. P., Blok, H. E., & Feng, L. (Eds.). (2007). *Multimedia retrieval*. Springer Science & Business Media.
4. Scripts, presentations and other learning material available in the e-course.

[37] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

11. Optional/additional literature (at the time of submission of the study programme proposal)		
12. Number of assigned reading copies in relation to the number of students currently attending the course		
Title	Number of copies	Number of students
13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences		
Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).		

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Martina Holenko Dlab, PhD	
Name of the course	Computer Graphics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for MMS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about basic concepts and techniques for representing graphics objects (2D and 3D) and to gain skills in using software for the creation of object models and their representation on a computer.

2. Course enrolment requirements

A pass mark in Multimedia Systems.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe basic concepts of computer graphics, and of modelling and representation processes for 2D and 3D objects.
I2. Create raster and vector graphics representations of given objects.
I3. Apply procedures for determining hidden lines and surfaces on a graphical representation.
I4. Apply colouring procedures, simple lighting and shading models for a graphical representation.
I5. Analyse the mode of operation of basic computer graphics algorithms.
I6. Choose an appropriate software for modelling 2D and 3D objects and their representation on a computer and in the web browser.

4. Course content

The course includes the following topics:

- Basic concepts of computer graphics and fields of application.
- Mathematical bases for computer graphics.
- Raster graphics and rasterization algorithms.
- Object modelling and representation.
- Projection of objects in 3D space and view transformations.
- Determining visible and hidden lines and surfaces.
- Colours in computer graphics.
- Models and procedures of lighting and shading.
- Textures.
- Preparing 2D and 3D graphics for viewing on a computer and in the web browser.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network

	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining auditory classroom work (lectures and part of exercises), computer laboratory work (exercises), individual work outside the classroom (homework and project assignment) and distance learning by using the chosen e-learning system. A detailed schedule and topics of lectures and exercises will be specified in the course syllabus. When they enrol into this course, students will be instructed to use the distance learning system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Actively participate in solving practical assignments given in class and for homework.
- Participate in continuous assessment (preliminary exams) and successfully pass them.
- Create and continuously update a portfolio consisting of graphical contents created during the course.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[38] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1
Portfolio	0.5						

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts related to the basics of computer graphics, e.g. through multiple choice questions, fill in the blank questions and essay questions – I1, I3, I4, I5.
- Homework with practical assignments in which students show their understanding of theoretical and practical concepts related to the creation of simple 2D and 3D object representations. For doing their homework, students need to use an appropriate software (e.g. Adobe Photoshop, Illustrator, Blender, Autodesk 3DS Max, Maya) – I2, I3, I4, I6.
- Project assignment in which students demonstrate their understanding of the whole theoretical and practical course material. In this process, students model 2D and 3D representations of chosen objects from the real world and their transformations, including object lighting, shading, colouring etc. While working on their project, students document their own work in written form and use an

[38] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

appropriate software (e.g. Adobe Photoshop, Illustrator, Blender, Autodesk 3DS Max, Maya). The lecturer evaluates the project assignment according to the criteria set in advance – I1, I2, I3, I4, I5, I6.

- A portfolio consisting of graphical contents created during the course. Completeness and quality of contents in the portfolio will be evaluated according to the criteria set in advance – I2, I3, I4, I6.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Angel, E. Shreiner, D.: Interactive Computer Graphics: A Top-Down Approach with WebGL (7th Edition), Pearson Education, Inc., publishing, 2015.
2. Marschner, S. & Shirley, P.: Fundamentals of computer graphics (4th edition). A K Peters / CRC Press, 2015.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Hughes, F. J. et al. (2014). Computer graphics: principles and practice (3rd edition). Upper Saddle River, NJ: Addison-Wesley.
2. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Nataša Hoić-Božić, PhD / Vanja Slavuj, PhD	
Name of the course	Computer Animation	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for MMS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about techniques and methods of 2D and 3D animation from the field of computer graphics, and to gain and develop skills for applying the appropriate software to animate characters and inanimate objects.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Distinguish between different animation types and basic animation techniques and principles.

I2. Apply basic geometric transformations in two-dimensional and three-dimensional space by using the chosen software.

I3. Perform rigging.

I4. Recognize basic requirements of animation timing on specific examples.

I5. Plan a complete virtual scene for animation, including the choice of an appropriate camera model and appropriate lighting.

I6. Create and present an interactive 3D (or hybrid) animation, prepared according to own production plan, by using the appropriate software.

4. Course content

The course includes the following topics:

- Introduction to animation – historical development and fields of application
- 2D and 3D geometric transformations
- 2D and 3D animation techniques and basic animation principles
- Storyboarding
- Rigging
- Timing
- Camera model, lighting model, scene planning
- Rendering
- Animation and interaction

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network

	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining auditory classroom work (lectures), computer laboratory work (exercises), individual work outside the classroom (homework and project assignment) and distance learning by using the chosen e-learning system. A detailed schedule and topics of lectures and exercises will be specified in the course syllabus. When they enrol into this course, students will be instructed to use the distance learning system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures and auditory and/or laboratory exercises;
- Actively participate in practical problem-solving during lectures and auditory and/or laboratory exercises;
- Participate in continuous assessment (theoretical preliminary exams) and successfully pass them;
- Create and continuously update a portfolio consisting of contents created during the course;
- Complete a project assignment on a given topic (practical work instead of written exam), in pairs or in teams, and present it to the course lecturer/assistant in the oral exam. Both activities require at least 50% of the anticipated total number of credits.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[39] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam	0.5	Essay		Research	
Project	1	Continuous assessment	1.5	Report		Practical work	
Portfolio	0.5						

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written assessment (theoretical preliminary exam) in which students demonstrate their understanding of the basic theoretical concepts related to computer animation (e.g. performing geometric transformations and their mathematical basis, properties of basic animation techniques, rigging, basic concepts of scene planning etc.), and which can include remembering and fill in the gap questions, multiple choice questions, matching questions, essay questions and extended response questions – I1, I3, I4, I5.
- Portfolio in which students include their own practical works created during exercises and at home (as part of homework). The portfolio includes solutions to short, simple practical assignments in which students demonstrate their understanding of theoretical and practical concepts related to 2D and 3D animation, and their skills necessary for 2D/3D representation and object and character animation on a computer. During the exercises and when doing their homework, students need to

[39] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

use the appropriate software (e.g. Adobe Character Animator, Autodesk 3DS Max, Autodesk Maya, Blender) – I2, I3, I4, I5.

- Creation of a project assignment (instead of written exam), in groups or in pairs, in which students demonstrate their understanding of the entire theoretical and practical course material and integrate it in the practical work. In this process, students model medium complexity 2D and 3D objects and characters and animate them, taking into account, among other things, framing rules, scene lighting methods and animation timing. While working on their project, students need to extensively document their own work in written form and use the appropriate software. The lecturer evaluates their work by using the criteria known to students before starting the project – I2, I3, I4, I5, I6.
- The oral exam is related to the project assignment, and students are expected to provide valid reasons for the decisions they made in doing their project assignment, through the presentation of their work and conversation with the lecturer. The lecturer evaluates the presentation and conversation with students using the criteria known to students before creating the presentation and engaging in the conversation – I6.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Beane, A. (2012). 3D animation essentials. Indianapolis, IN: John Wiley & Sons, Inc.
2. O'Hailey, T. (2015). Hybrid animation: Integrating 2D and 3D assets. Burlington, MA: Focal Press.
3. Roberts, S. (2013). Character animation fundamentals: Developing skills for 2D and 3D character animation. Burlington, MA: Focal Press.
4. Williams, R. (2012). The animator's survival kit (Expanded edition). New York: Farrar, Straus and Giroux.
5. Appropriate software manuals.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Chopine, A. (2011). 3D art essentials. Burlington, MA: Focal Press.
2. Halas, J. & Whitaker, H. (2009). Timing for animation (2nd edition). Burlington, MA: Focal Press.
3. Pandžić et al. (2011). Virtualna okruženja: Interaktivna 3D grafika i njene primjene. Zagreb: Element.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Patrizia Poščić, PhD	
Name of the course	Database Administration and Security	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for IS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master basic knowledge from the field of database administration and security. This knowledge includes, but is not limited to, creation of database environment, change management and performance management, ensuring database security, setting up security strategies and other DBA tasks. Furthermore, the objective of the course is to work with procedural and non-procedural query languages and to create a business application based on databases.

2. Course enrolment requirements

A pass mark in Databases.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

- I1. Explain basic concepts from the field of database administration.
- I2. Distinguish between different types of database administrators and their basic responsibilities.
- I3. Create and maintain database development environment by using appropriate DBA tools.
- I4. Apply appropriate methods for database backup and recovery.
- I5. Compare different techniques for ensuring database security.
- I6. Choose an appropriate database security strategy and achieve a satisfactory security level.
- I7. By using procedural and non-procedural query languages, create simple and complex program blocks over a relational database.
- I8. Within a given programming environment, create a software solution based on a relational database.

4. Course content

Database administration. DBA tasks. Types of DBAs. Creating a database environment. Change management. Data availability. Performance management. Backup and recovery. DBA tools.

Database security. Techniques for ensuring database security. Keeping sensitive data secure. Security strategies.

Business applications based on relational databases. RAD tools. Procedural and non-procedural query languages for working with a relational database.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network

	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes, actively participate in all course activities and follow course activities within the distance learning system.
- Participate in continuous assessment (preliminary exams and quizzes) and achieve the number of credits equal to or higher than the passing score (if any).
- Complete a project assignment (practical work – application creation) on a given topic and score at least 50% of credits.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[40] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	2	Report		Practical work	0.5
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through activities in class, continuous assessment (preliminary exams and quizzes) and a practical project assignment, in addition to computer work.

- In the theoretical assessment, students demonstrate their understanding of theoretical concepts from the field of database administration (I1, I2), they compare different techniques for ensuring database security (I4, I5) and distinguish between different database security strategies (I6). E.g. describe at least three types of database administrators and their tasks.
- During the computer laboratory work, students actively create and maintain a database environment by using appropriate DBA tools and apply appropriate database backup and recovery methods (I3, I4). E.g. adjust a database environment in SQL Developer tool.
- In the practical preliminary exam, students independently build simple and complex program blocks over a relational database on a computer, according to the set requirements (I7). E.g. create a procedure for updating worker addresses in the database.

[40] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- In the project assignment, students create a software solution (business application) with a user interface in a given tool, based on the previously created database (I8). E.g. create a business application, including model and database, for library operations.

10. Mandatory literature (at the time of submission of study programme proposal)

1. S. Balamurugan, S. Charanyaa (2014). Principles of Database Security. OmniScriptum.
2. C. S. Mullins (2013). Database Administration: the Complete Guide to DBA Practices and Procedures. Addison-Wesley.
3. A. Geller, B. Spendolini (2017). Oracle Application Express (APEX): Build Powerful Data-Centric Web Apps with APEX. McGraw-Hill.
4. Poščić, P. (2018). Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. E. Sciore (2015). Understanding Oracle APEX 5 Application Development. Apress.
2. J. Murach (2014). Murach's Oracle SQL and PL/SQL for developers. Mike Murach & Associates.
3. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Sanja Čandrlić, PhD	
Name of the course	User Interface and Interaction Design	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for IS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to basic concepts, methods and techniques of user interface, user experience and interaction design.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse tasks/requirements on which user interface and interaction design is based.

I2. Create a persona and scenarios for navigating through the application.

I3. Create a simple user interface prototype.

I4. Organize the content according to device requirements.

I5. Apply appropriate elements in the user interface.

4. Course content

- User interface (UI). User experience (UX). Interaction design (IxD).
- User. Cognitive principles, attention, perception, recognition, memory. Mental models, mapping, metaphors. Design patterns.
- User-centered design (UCD). Qualitative user research. User modeling. Personas and goals. Scenarios and requirements. From requirements to design. User interface prototype. Validation and usability testing.
- Different platforms for interactive products. Context for interactive system design.
- Visual interface design. Principles of visual interface design. Multimodal interface design. Information design. Principles of visual information design.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work,	

	individual work outside the classroom and e-learning.

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment and successfully pass it.
- Undertake practical problem-solving tasks and successfully complete them.
- Make a project, independently or in groups, and present it to the lecturer.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[41] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1.5	Continuous assessment		Report		Practical work	1
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of concepts from the field of user interface, user experience and interaction design (I1, I2, I4), e.g. through multiple choice questions, fill in the blank questions and essay questions students name the advantages of using a persona in the design process, list key components for the implementation of formative usability testing and explain principles of visual interface design.
- Practical assignments in which students create a persona and scenarios (I2) and organize the content. (I4)
- Project in which students, individually or in teams, need to work out an interaction with the system. (I2, I3, I5)

10. Mandatory literature (at the time of submission of study programme proposal)

1. Alan Cooper, Robert Reimann, David Cronin & Chris Noessel. About Face – The Essentials of Interaction Design. Wiley Publishing, 2014.
2. Jenifer Tidwell. Designing Interfaces – Patterns for Effective Interaction Design. O'Reilly, 2011.
3. David Benyon. Designing Interactive Systems. Pearson, 2014.
4. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Jenny Preece, Yvonne Rogers & Helen Sharp. Interaction Design: Beyond Human-Computer Interaction, John Wiley and Sons, 2019.
2. Cennydd Bowles & James Box. Undercover User Experience Design. Peachpit, Pearson Education, 2011.

[41] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

3. Jeff Johnson. Designing with the Mind in Mind. Elsevier, 2014.
4. Soren Lauesen. User Interface Design: A Software Engineering Perspective. Addison-Wesley, 2004.
5. Steve Krug. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. New Riders, Peachpit, Person Education, 2014.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Sanja Čandrlić, PhD / Martina Ašenbrener Katić, PhD	
Name of the course	Information systems for specific purposes	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for IS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to different information systems for specific purposes.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain basic concepts related to specific types of information systems.

I2. Recognize type and characteristics of specific purpose information systems, e.g. information systems used for generating reports, extracting data, helping make decisions etc.

I3. Analyse different expert systems.

I4. Apply tools to create a database for office management.

I5. Apply different open-source information systems.

4. Course content

- Classifications, types and elements of information systems. Goals and levels of information system management. Decision support systems. Executive information systems. Accounting information systems. Office automation systems.
- Business information systems (ERP). Advantages of ERP. Components of ERP. Phases of ERP life cycle. ERP products.
- Expert systems. Historical overview. Overview of various expert systems from the field of finance, medicine, manufacturing, accounting, process control, certain activities etc. Knowledge-based systems.
- Management information systems. Historical overview. Advantages. Types.
- Geographic information systems (GIS). Concept of GIS. Definition. Application of GIS. Components of GIS. GIS projects in Croatia and worldwide.
- Open-source information systems. Overview of different open-source information systems (e.g. decision support systems, executive IS, accounting IS, ERP, GIS etc.)

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network

	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment and achieve the number of credits equal to or higher than the passing score (if any).
- Participate in practical problem-solving tasks and achieve the number of credits equal to or higher than the passing score (if any).
- Individually or in teams, make a project and present it to the lecturer, and achieve the number of credits equal to or higher than the passing score (if any).
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[42] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	0.5	Oral exam		Essay		Research	
Project	1.5	Continuous assessment	1	Report		Practical work	0.5
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts related to different information systems (I1, I2). For example, through multiple choice questions, fill in the blank questions and essay questions students name individual components and functions of each system.
- A project in which students, individually or in teams, need to choose an expert system (e.g. MYCIN) and analyse it (I3). Students present their solution to the lecturer. For example, students analyse the MYCIN system individually or in teams and present the results to other students in front of the lecturer.
- A project in which students, individually or in teams, need to apply tools for creating office databases (I4) (for example, create tables, queries, masks, reports etc. using one of the chosen tools

[42] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

for creating databases, e.g. MS Access tool)

- Practical problem-solving task in which students, individually or in teams, apply one of the open-source information systems (I5). For example, according to the assignment requests, students create a solution to a task, individually or in teams, using a chosen open-source information system (e.g. accounting operating system) and present it to other students and the lecturer.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Scripts, presentations and other learning material available in the e-course.
2. Bourgeois, D. T.; Information Systems for Business and Beyond; The Saylor Foundation's Open Textbook Challenge, 2014.
3. Bolstad, P.; GIS Fundamentals: A First Text on Geographic Information Systems, Fifth Edition; XanEdu Publishing Inc; 2016.
4. Giarratano, J.C.; Riley, G.D. Expert Systems: Principles and Programming. 4th ed.; PWS Publishing Company, Boston, 2004.
5. Laudon, K. C.; Laudon, J. P.; Management Information Systems: Managing the Digital Firm (15th Edition); Pearson; 2017.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Pavlić, M., Informacijski sustavi, Školska knjiga, Zagreb, 2011.
2. Luger, F.G.; Stubblefield, W.A. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 6th ed., Addison-Wesley, Harlow, 2009.
3. Romney , M. B.; Steinbart, P. J.; Accounting Information Systems (14th Edition); Pearson; 2017
4. Longley, P.A.; Goodchild, M.F.; Maguire, D.J.; Rhind D.W. Geographic Information Systems and Science. 2nd Edition, Wiley, 2005.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION

Course instructor	Asst. Prof. Marija Brkić Bakarić, PhD	
Name of the course	System Dynamics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about system-based approach and methods and apply system dynamics to predict the behaviour of simple systems.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students will be able to:

I1. Apply system-based approach and black box method.

I2. Identify feedback loops and determine types of feedback loops in the system.

I3. Create models of simple dynamic systems by applying system dynamics principles.

I4. Predict system behaviour based on system description.

I5. Use network methods to solve problems.

4. Course content

Concept of system. Basic properties of a system. Complex system. Models and modelling. System-based approach. Systems analysis. Black box method. System dynamics. Feedback loops. Modelling and predicting system behaviour. Network methods for temporal system analysis.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	E-learning system will be used in this course. When they enrol into the course, students will be instructed to use the tools available in the system. A detailed schedule will be provided in the syllabus.	

7. Student responsibilities

Students should regularly participate in all course activities (preliminary exams, projects, problem-solving tasks, homework) and score at least 50% on the final exam. A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[43] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam	0.5	Essay		Research	
Project	0.5	Continuous assessment	1.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

In the written or oral work, students, based on a given system description, demonstrate their knowledge of the system-based approach and black-box approach in practice (I1), e.g. based on a record obtained through a black-box approach, establish a relationship between inputs and outputs.

In the written or oral work, students analyse and determine the dominant feedback loop in the system based on a given system description (I2), e.g. determine loop behaviour and identify the dominant loop based on the system description provided.

Practical assessment (practical preliminary exam) in which students, based on the provided description of a real dynamic system, create a system dynamics diagram using a computer program of their choice, and formalize the system behaviour by executing appropriate mathematical equations (I3), e.g. create a stock and flow diagram with two state variables for modelling students and faculty staff if it is known that the higher ratio makes the university less attractive and at the same time leads to employment. Initial values are provided.

Group or individual practical project assignment in which students, according to the set instructions and evaluation criteria, analyse the chosen real dynamic system, and based on the analysis and data collected, create a system dynamics diagram using the computer program of their choice and formalize the system behaviour by entering and executing appropriate mathematical equations. By changing individual parameters or model structure, students simulate changes in the system and predict system behaviour (or behaviour of its individual components) over time (I3), e.g. create a model of the demographic picture of the Republic of Croatia. Propose at least two measures for demographic recovery and simulate system behaviour.

In their written or oral work, students predict system behaviour based on the system description provided (I4), e.g. choose a chart of system behaviour based on the system description provided.

Practical assessment (practical preliminary exam) in which students apply the network method of their choice to calculate the critical path and system time optimization (I5), e.g. based on the system description provided, calculate the duration of critical path and determine the element on which we need to act to reduce the duration of critical path.

10. Mandatory literature (at the time of submission of study programme proposal)

[43] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

1. Scripts, presentations and other learning material available in the e-course.
2. Sterman, J. H: Business dynamics. Irwin McGraw-Hill: 2000.
3. Morecroft, J. D. W.: Strategic modelling and business dynamics. John Wiley and Sons Ltd.: 2015.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Duggan, J.: System Dynamics Modeling with R. Springer: 2016.
2. Juan Martin Garcia. Theory and Practical exercises of System dynamics. 2006.
3. Radošević, D. Osnove teorije sustava. Zagreb, Nakladni zavod Matice hrvatske: 2001.
4. Draper L. Kauffman, Jr. Systems One: An introduction to systems thinking. Future Systems: 1980.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Maja Matetić, PhD	
Name of the course	Programming for Data Science	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

Data science deals with data collection, preparation and analysis, interpretation of analysis results and visualization of big and complex data sets. Skills required from a data scientist include the preparation of raw data, exploratory data analysis, machine learning and interpretation of data analysis results. The basis for these skills consists of computer programming with a rich software ecosystem for data science. The objective of this course is to ensure high-quality foundations for programming for data science, through creation of a program using current programming languages suitable for programming for data science (e.g. Python and R).

2. Course enrolment requirements

A pass mark in Object-Oriented Programming.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students will be able to:

I1. Apply procedures for loading, integrating and transforming data into a suitable form for data analysis.

I2. Apply exploratory data analysis.

I3. Explain, implement and apply chosen algorithms for supervised and unsupervised machine learning.

I4. Explain, implement and use evaluation procedures and efficiency assessment for supervised and unsupervised machine learning.

I5. Apply visualization procedures.

I6. Interpret the results of data analysis.

4. Course content

- Syntax and semantics of programming languages that are particularly suitable for programming for data science, for example Python, R.
- Procedures for loading, combining, transforming and choosing data.
- Exploratory data analysis.
- Algorithms for solving missing value problems, discretization and dimensionality reduction.
- Algorithms for supervised machine learning, e.g. Naive Bayes, decision trees, random forests
- Algorithms for unsupervised machine learning, e.g. k-means clustering.
- Libraries for data analysis.

- Evaluation procedures and metrics for model quality assessment.
- Visualization and analysis of results.
- Data analysis is performed on chosen, publicly available datasets from different fields and of different types (for example: business and financial data, sensor data from mobile phones, medical data, textual data, social network data).

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	□ multimedia and network
	☒ exercises	□ laboratories
	□ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	E-learning system will be used in this course. A detailed schedule will be provided in the syllabus.	

7. Student responsibilities

Students should regularly participate in all course activities (seminar paper, project, exercises, preliminary exam) and score at least 50% on the final exam. A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[44] of student work

Class attendance	1	Class participation	0.5	Seminar paper	0.5	Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1.5	Continuous assessment	1.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

Exercises are carried out on a computer, and students solve tasks related to exploratory data analysis (e.g., summing up data and visualization of relevant predictors from a data set), preparation of data (e.g. discretization and solving missing value problems) and data analysis (I1, I2, I3), e.g. by performing a required clustering procedure for a data set.
Creation of seminar paper, in which students address a machine learning algorithm of their choice and present their work individually or in teams. Content relevance with respect to the given topic is evaluated, as well as the originality of ideas presented by the student in his/her seminar (I3, I4), e.g. the method of preparing the data, choice of procedure parameters and interpretation of the model obtained through learning.

Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of algorithms for supervised and unsupervised machine learning and of evaluation procedures for data set model assessment (I3, I4). Assessment can be performed through single-answer questions or fill in the blank questions.

Creation of project assignment for a given data set, which includes the preparation of data for analysis, choice of machine learning procedure, conducting the experiment (data analysis), representation and interpretation of results (I1-I6). For example, build and interpret descriptive and predictive models (e.g. regular and irregular heartbeat) for a set of medical sensor data (heart rate etc.) collected from the mobile

[44] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

device by applying machine learning procedures.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Shmueli, Galit, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, and Kenneth C. Lichtendahl Jr. Data mining for business analytics: concepts, techniques, and applications in R. John Wiley & Sons, 2018.
2. J. VanderPlas, Python Data Science Handbook: Essential tools for working with data (1st ed.), O'Reilly Media Inc., 2017.
3. Baumer, Benjamin S., Daniel T. Kaplan, and Nicholas J. Horton. Modern data science with R. CRC Press, 2017.
4. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Python Data Science Essentials: A practitioner's guide covering essential data science principles, tools, and techniques, 3rd Ed., Packt Publishing, 2018.
2. I. Witten, E. Frank, M. Hall and C. Pal, Data Mining: Practical Machine Learning Tools and Techniques (4th ed.), Morgan Kaufmann, 2016.
3. Roiger, Richard J. Data mining: a tutorial-based primer. Chapman and Hall/CRC, 2017.
4. Larose, Daniel T., and Chantal D. Larose. Discovering knowledge in data: an introduction to data mining. John Wiley & Sons, 2014.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Sanda Martinčić-Ipšić, PhD	
Name of the course	Introduction to Theoretical Computing	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about theoretical foundations of computing, finite-state automata, formal grammars, regular expressions, Turing machines and time and space complexity.

2. Course enrolment requirements

A pass mark in Algorithms and Data Structures.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Model a given regular expression by using finite-state automata.

I2. Perform transformations between different finite-state automata (deterministic, non-deterministic, with output etc.).

I3. Simplify formal grammar and find its normal form and transform formal grammar into appropriate automaton.

I4. Examine whether a given language is regular, context-dependent or context-free.

I5. Transform pushdown automata accepting by final state into pushdown automata accepting by empty stack and vice versa and construct a pushdown automaton for context-free grammar.

I6. Construct a Turing machine for a given recursive or recursively enumerable language and for a given computing problem.

I7. Work out a solution to a given problem (communication protocols, digital circuits, regular word form etc.) using formal models of finite-state automata or grammars.

4. Course content

- Introduction to theoretical computing.
- Finite-state automata: deterministic and non-deterministic finite-state automata, automata with e-transitions.
- Automata with output.
- Regular expressions, transformation to nDFA, pumping lemma.
- Regular grammars, simplification of grammars.
- Chomsky and Greibach normal forms.

- Parsers. Parse tree. Top-down parsing, recursive descent, bottom-up.
- Pushdown automaton. Context-free languages. Pumping lemma.
- Turing machine. Recursive and recursively enumerable languages. Computability and decidability.
- Context-dependent languages. Chomsky hierarchy.
- Time and space complexity.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	☒ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow and take part in course activities.
- Participate in continuous assessment (preliminary exams).
- Do homework and complete individual assignments during exercises.
- Write a seminar or program paper that includes the use of elements listed under learning outcomes. (In the practical work (written seminar paper and/or code), students need to work out and realize a solution to a given problem (communication protocols, digital circuits, regular word forms, parsing etc.) and/or work out and realize a solution to a given problem in the form of a Turing machine.
- Written (or online) final exam for the course covers all the course material and requires a score of 50% to pass.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[45] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1.5	Continuous assessment	1	Report		Practical work	
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

[45] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

The set of learning outcomes for the course Introduction to Automata Theory is assessed through written/oral/online examination and practical work. Online assessment is performed under controlled conditions.

- Written/online assessment is used to test the knowledge of transforming a regular expression into an automaton, and of transforming between different finite-state automata (deterministic, non-deterministic, with output etc.). (I1, I2)
- Written/online assessment is used to test the knowledge of simplifying grammar and writing in normal form, as well as of transforming it into an appropriate finite-state automaton. (I3)
- Written/online assessment is used to test the application of theoretical knowledge to prove the properties of regular, context-dependent and context-free languages. (I4)
- Written/online assessment is used to test the knowledge of transforming pushdown automata accepting by final state into pushdown automata accepting by empty stack and vice versa and of designing a pushdown automaton for context-free grammar. (I5)
- Written/online assessment is used to test the knowledge of the construction of a Turing machine for a given recursive or recursively enumerable language. (I6)
- In the practical work (written seminar paper and/or code), students need to work out and realize a solution to a given problem (communication protocols, digital circuits, regular word forms, parsing etc.). (I7, I1, I2, I3, I4, I5, I6)
- In the practical work (written seminar paper and/or code), students need to work out and realize a solution to a given problem in the form of a Turing machine (I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. S. Srbljić. Uvod u teoriju računarstva, Element, Zagreb, 2010.
2. Learning material available in the e-learning system, together with own notes and materials from lectures and exercises.
3. M. Spiser, Introduction to the Theory of Computation, Cengage learning, 3rd edition, 2013.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. J. E. Hopcroft, R. Motwani, J. D. Ullman. Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 3rd edition, 2001.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Sanja Rukavina, PhD	
Name of the course	Combinatorics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The main objective of the course is to introduce students to combinatorial thinking and combinatorial argument. For this purpose, during this course students need to:

- describe and compare different forms of Dirichlet's principle and generalization,
- analyse basic principles of enumerating elements in a finite set and combinatorial enumeration,
- define binomial and multinomial coefficients and analyse their properties,
- define multiplicative functions and analyse examples of multiplicative functions,
- define and distinguish between certain recursive problems and analyse the methods of resolving such problems,
- define and compare some combinatorial structures.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After taking the course and passing the exam, students will be able to:

I1. analyse and distinguish between applications of different enumeration methods or forms of Dirichlet's principle (A5, B6, C6, D6, E4, F5),
I2. provide valid arguments for choosing an enumeration method or a form of Dirichlet's principle, and apply an appropriate procedure when solving tasks (A5, B6, C5, D5, E4, F5),
I3. describe multiplicative functions and analyse examples of multiplicative functions (A4, B5, C5, D5, E4, F5),
I4. analyse recursive problems when solving combinatorial tasks by using well-argued procedures (A5, B6, C5, D5, E4, F5),
I5. provide valid arguments for the use of specific properties of binomial and multinomial coefficients when solving tasks (A5, B6, C5, D5, E4, F5)
I6. formulate combinatorial interpretations of expressions to substantiate different statements (A6, B6, C6, D6, E4, F5)
I7. describe some combinatorial structures (A4, B5, C5, D5, E4, F5)
I8. mathematically prove the foundation of all procedures and formulas used within this course (A6, B6, C6, D6, E4, F5).

4. Course content

Basic principles of enumeration. Dirichlet's principle. Ramsey’s theorem. Permutations and combinations of sets and multisets. Binomial and multinomial coefficients. Inclusion-exclusion formula. Multiplicative functions. Recursive relations. Generating functions. Some combinatorial structures.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☒ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments		

7. Student responsibilities

Students must attend the classes, actively participate in all forms of classes, achieve a certain number of credits during the semester and pass the final exam (details will be provided in the course syllabus).

8. Monitoring[46] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam	1.5	Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through short online tests, homework, preliminary exams and an oral exam.

- In online tests, students demonstrate their understanding of theoretical concepts related to the course material (I1, I2, I3, I4, I5, I6, I7, I8) by answering to essay questions, multiple choice questions, and circling, gap filling and matching questions. For example, under which assumption a simple form can be derived from the strong form of Dirichlet's principle?
- In their homework, students demonstrate that they have mastered the course material by applying their knowledge to specific tasks, analysing the properties and choosing appropriate methods to solve problems (I1, I2, I3, I4, I5, I6, I7, I8). For example, determine a recurrence relation and corresponding initial conditions, and determine its solution.
- In the written assessment (preliminary exam), students demonstrate that they have mastered the course material by applying their knowledge to specific tasks, analysing the properties and choosing appropriate methods to solve problems (I1, I2, I3, I4, I5, I6, I7, I8). For example, prove a given identity using combinatorics.
- In the oral assessment (final exam), students demonstrate that they have mastered the theoretical concepts from the course material by defining certain concepts, expressing and proving relevant properties and results. For example, define the concept of permutation of a set and prove a corresponding theorem on the number of permutations of a finite set (I1, I2, I3, I4, I5, I6, I7, I8).

10. Mandatory literature (at the time of submission of study programme proposal)

[46] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

1. D. Veljan: Kombinatorika i diskretna matematika, Algoritam, Zagreb, 2001.
2. M. Cvitković, Kombinatorika, zbirka zadataka, Element, Zagreb, 2001.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. D. Žubrinić, Diskretna matematika, Element, Zagreb, 1997.
2. D. Veljan, Kombinatorika s teorijom grafova, Školska knjiga, Zagreb, 1989.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Bojan Crnković, PhD	
Name of the course	Numerical Mathematics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to the basic methods of numerical mathematics and the application of such methods to specific problems. For this purpose, during this course students need to:

- describe and distinguish between different numerical errors,
- describe different methods of software optimization used in the course of numerical computation,
- describe problems to be solved using numerical methods for function interpolation,
- define and distinguish between different interpolation polynomials, analyse their properties and apply them to specific problems,
- describe the method for polynomial function interpolation, analyse the properties of such interpolation and apply them to specific problems,
- determine errors of interpolation polynomials and advantages and disadvantages of each processed interpolation polynomial,
- describe numerical methods for solving non-linear equations, analyse their properties and apply methods to specific problems,
- describe numerical methods for solving algebraic equations, analyse their properties and apply methods to specific problems,
- describe numerical methods for calculating definite integrals, analyse their properties and apply methods to specific problems,
- describe numerical methods for solving ordinary differential equations, analyse their properties and apply methods to specific problems,
- demonstrate the use of modern computers for the purpose of solving different problems from the field of numerical mathematics, especially the problems presented during the course and point to the specific optimization of programmer algorithms with the aim of obtaining as accurate results as possible.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After taking the course and passing the exam, students will be able to:

I1. Critically analyse the problem of interpolation approximation for a given function (A7, B6, C5, D5, E4, F5).

I2. Apply one of the interpolation polynomials in a well-argued manner (A6, B6, C6, D5, E4, F5).

I3. Analyse the results obtained using interpolation methods and estimate errors that occur (A6, B5, C6, D5, E4, F5).

I4. Distinguish between polynomial interpolation and piecewise polynomial function interpolation and Analyse advantages and disadvantages of each method (A6, B4, C6, D5, E4, F5).

I5. Apply linear and cubic spline interpolation to a given problem and analyse the results obtained in a well-argued manner (A6, B4, C7, D5, E4, F5).

I6. Apply methods for approximate solution to algebraic and non-algebraic equations in a well-argued manner and estimate errors that occur (A6, B6, C5, D5, E4, F5).

I7. Apply methods of approximate integration to solve definite integrals and provide valid arguments for this (A6, B4, C7, D5, E4, F5).

I8. Distinguish between different methods for solving ordinary differential equations and apply methods for solving ordinary differential equations to specific problems and analyse the results obtained in a well-argued manner (A6, B4, C6, D5, E4, F5).

I9. Apply methods addressed in the course to solve specific problems by applying programs and using modern computers (A6, B6, C5, D5, E4, F5).

I10. Mathematically prove the foundation of all procedures and formulas used within this course (A6, B4, C6, D5, E4, F5).

4. Course content

Errors and type of errors. Round-off errors. Finite-field arithmetic. Solving systems of linear equations: condition of a system of linear equations, solving a triangular system, Gaussian elimination method, LU decomposition. Interpolation. Interpolation polynomials: Newton, Lagrange, Chebyshev and cubic spline interpolation polynomial. Error estimate and convergence. Determining zeros of real functions: iteration method, Newton method, secant method. Error estimate. Determining zeros of polynomials. Numerical integration: Newton-Cotes formulas, trapezoidal and Simpson's formula. Convergence and error estimate. Numerical solving of ordinary differential equations: Euler's method, Runge-Kutta method, discretization method. Stability of numerical algorithms on modern computers.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	☒ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Exercises in this course will be performed in auditory form (10 class hours) and on computers (20 class hours).	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes and participate in course activities within the distance learning system.
- Participate in assessments (tests and preliminary exams) and obtain enough credits to take the exam. Minimum number of credits (per activity) necessary to take the final exam is specified in the detailed course syllabus.

- In the final exam, present the acquired material in a well-argued manner according to the content of the course. Passing score for the exam is 50%.

A detailed scoring system for the course, including individual passing scores, will be specified in the course syllabus.

8. Monitoring[47] of student work

Class attendance	1	Class participation	0.5	Seminar paper	0.5	Experimental work	
Written exam	1	Oral exam	1	Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through short tests, preliminary exams and an oral exam.

- In the written or online assessment (short tests), students demonstrate the basic understanding of theoretical concepts, distinguishing between defined concepts from the course material (I1-I8), e.g. through multiple choice questions, fill in the blank questions and essay questions.
- In the written assessment (preliminary exams), students demonstrate that they have mastered the course material through mathematical modelling of problems and by applying numerical methods to specific tasks using the computer (I9). For example, find an approximate solution to a definite integral with a given relative accuracy.
- In the written and oral assessment (final exam), students demonstrate that they have mastered the theoretical concepts of the course material by formulating mathematical statements, analysing properties and discussing examples, and proving the statements (I1-I8, I10). For example, derive a trapezoidal formula and express and prove the error estimate theorem for the formula.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Rudolf Scitovski, Numerička matematika, Elektotehnički fakultet, Osijek 1999.
2. J. Stoer, R. Bulirsch: Introduction to Numerical Analysis, second edition, Springer-Verlag, New York, 1991.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Ivan Ivanšić: Numerička matematika, Element, Zagreb, 1999.
2. M. Rogina, S. Singer, S. Singer: Numerička matematika, Zagreb 2002. (online)

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

[47] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Tajana Ban Kirigin, PhD	
Name of the course	Mathematical Logic	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic concepts of logic. Logical reasoning is the basis for entire mathematics. This course enables students to:

- get acquainted with formal mathematical logic
- get acquainted with the division into syntax and semantics and their relationship through the basics of propositional logic and first-order logic
- though formal logic systems, the understanding of rigorous proof and theorem is specified, as well as the understanding of mathematical objects.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After taking the course and passing the exam, students will be able to:

I1. describe the concept of formula and formulate the outcome of the formula in propositional logic and first-order logic (A6, B6, D5, E5, F6)

I2. formulate the concepts of validity, satisfiability and refutability of a formula and analyse them on examples, as well as implication and equivalence of propositional logic and first-order logic formulas (A6, B6, D5, E5, F6)

I3. analyse the representation of propositional connectives (A6, B6, D5, E5, F6)

I4. construct normal forms for propositional logic formulas (A6, B6, D5, E5, F6)

I5. construct prenex normal forms for first-order logic formulas (A6, B6, D5, E5, F6)

I6. formulate the concept of proof and the concept of theorem in formal systems of propositional calculus and natural deduction and describe basic corresponding meta-results (A6, B6, D5, E5, F6)

I7. analyse and construct deductions in the natural deduction system for propositional logic (A6, B6, D5, E5, F6)

I8. explain the role of mathematical logic in the whole of mathematics as a science, historical and intuitive importance of propositional logic and reasons why stronger logical theories have emerged, especially first-order logic (A6, B6, D5, E5, F6)

I9. describe basic meta-results and limitations of first-order logic (A5, B5, D5, E5, F6)

I10. mathematically prove the foundation of procedures and theoretical results used in this course (A6, B6, D5, E5, F5).

4. Course content

Conventional propositional logic: syntax, semantics. Propositional connectives. Conjunctive and disjunctive normal form. Craig's lemma. Validity tests.

Propositional calculus and natural deduction system: consistency and completeness of a set of formulas, deduction theorem, adequacy theorem, completeness theorem and compactness theorem. Limitations of propositional logic. First-order theories: syntax, semantics. Prenex normal forms. Main test for first-order logic. Basic meta-results and limitations of first-order logic.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments		

7. Student responsibilities

Students must attend the classes, actively participate in all forms of classes, achieve a certain number of credits during the semester and pass the final exam (details will be provided in the course syllabus).

8. Monitoring[48] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam	1.5	Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through short tests, preliminary exams and an oral exam.

- In the written assessment (tests), students demonstrate their understanding of theoretical concepts from the course material (I1, I2, I3, I4, I5, I7, I10) on specific tasks and through essay questions. For example, determine a perfect disjunctive normal form of a given formula and based on this form explain whether the formula is valid in a well-argued manner.
- In the written assessment (preliminary exam), students demonstrate that they have mastered the course material by applying their knowledge to specific tasks, analysing the properties and choosing a method or applying a set problem-solving method (I1, I2, I3, I4, I5, I7, I10). For example, determine whether the given first-order logic formulas are logically equivalent.
- In the written and oral assessment (final exam), students demonstrate that they have mastered the theoretical concepts from the course material by defining certain concepts, expressing and proving relevant properties and results. For example, define the concept of deduction in the natural deduction system for propositional logic and prove the corresponding adequacy theorem (I1, I2, I3, I4, I5, I6, I7, I8, I9, I10).

10. Mandatory literature (at the time of submission of study programme proposal)

[48] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

1. M. Vuković: Matematička logika, Element, 2009.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. A. G. Hamilton: Logic for Mathematicians, Cambridge, University Press, 1988.
2. E. Mendelson: Introduction to Mathematical Logic, Chapman and Hall, 1964.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Ana Meštrović, PhD	
Name of the course	Bachelor's Thesis	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	8
	Number of class hours (L+E+S)	

1. Course objectives

The objective of the course is to teach students how to independently solve a more complex problem-solving task by preparing a bachelor's thesis from the area of their choice. With their bachelor's thesis, students need to demonstrate that they have acquired competencies and achieved learning outcomes related to problem solving from the fields included in their studies. They also need to demonstrate that they know how to use theoretical and practical knowledge gained during their studies.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse a given problem from the field of informatics and model its solution.

I2. Identify appropriate methods, techniques and tools for solving a given problem or question interesting for and relevant to the field of informatics.

I3. Choose appropriate ICT tools to process and represent the data collected.

I4. Use appropriate ICT tools to shape their knowledge and express it in a creative manner.

I5. Plan activities and resources for solving a given problem.

I6. Implement the action plan for solving a given problem.

I7. Provide academically valid arguments to the set questions.

I8. Work out a strategy for searching scientific databases and other sources in order to find relevant sources of scientific and expert information (printed and digital collections) using library services and services available online.

I9. Question the reliability and quality of sources of expert and scientific information available online.

I10. Demonstrate the understanding of concepts such as plagiarism, self-plagiarism, quotation, referencing, paraphrasing.

I11. Apply guidelines for formatting professional papers and presentations.

I12. Organise the content of presentation in accordance with its concept, main ideas and academic arguments.

I13. Make a presentation plan with respect to available time and target audience.

4. Course content
The course includes the methodological and practical aspect of preparing the bachelor’s thesis, including the choice of topic, its elaboration and final formatting. Students are introduced to the solving of a more complex problem-solving task, working with targeted literature and sources. Problem area of the thesis can include a specific information and communication system, which will require field work. Practical aspects of preparing the thesis and methodological approach for algorithmic problem solving are specially addressed. Methods of citing sources and analysing professional and scientific papers are also tackled. Division of thesis into chapters is conceived, as well as the choice of topic and evaluation of thesis expertise. Ethical aspects of research are considered in correlation with generally accepted principles of academic life, expressed through codes of ethics, and with expected social implications of the topic.

5. Manner of instruction	□ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	□ exercises	□ laboratories
	☒ distance learning	☒ mentorship
	□ fieldwork	☒ other ____________________
6. Comments		

7. Student responsibilities
Students must choose a mentor and the topic of their bachelor’s thesis, and individually prepare both the practical and the professional and scientific part of the thesis, through mentorship and consultations. In addition, the course has been aligned with the provisions of the Ordinance on Bachelor’s Thesis at Undergraduate University Study Programmes of the Department of Informatics at the University of Rijeka, which prescribes the responsibilities of students regarding the creation and presentation of the bachelor’s thesis.
8. Monitoring[49] of student work

Class attendance		Class participation		Seminar paper	2	Experimental work	
Written exam		Oral exam	1	Essay		Research	3
Project		Continuous assessment		Report		Practical work	2
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)
Learning outcomes are assessed through continuous assessment in the course of mentorship, creation and presentation of the bachelor’s thesis. Final exam consists in the presentation of the bachelor's thesis, i.e. oral presentation of the results of the bachelor’s thesis, with a prepared presentation and assessment of knowledge from the field of the bachelor's thesis.
10. Mandatory literature (at the time of submission of study programme proposal)
1. Vujević, M. Uvod u znanstveni rad u području društvenih znanosti. Informator, Zagreb, 1990. 2. Etički kodeks Sveučilišta u Rijeci. Sveučilište u Rijeci, Rijeka, 2003.
11. Optional/additional literature (at the time of submission of the study programme proposal)

[49] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Martina Holenko Dlab, PhD	
Name of the course	Undergraduate Internship	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	4
	Number of class hours (L+E+S)	

1. Course objectives

The objective of the course is for students to apply the competencies gained throughout their studies (knowledge, skills, independence and responsibility) in an actual work environment of the training provider.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Apply the acquired knowledge and skills in order to accurately, thoroughly and efficiently complete their work tasks in an actual working environment.

I2. Acquire knowledge and skills necessary to successfully complete their work tasks in an actual working environment.

I3. Analyse the appropriateness of tools, techniques and methods for completing their work tasks in an actual working environment.

I4. Behave in accordance with the instructions and feedback in the process of completing their work tasks in an actual working environment.

I5. Adapt to teamwork when completing their work tasks in an actual working environment.

I6. Adapt to business culture in an actual working environment.

4. Course content

The content of work tasks will depend on the profile of a work site (institutions, companies or other legal entities) where students will perform their internship.

5. Manner of instruction	□ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	□ exercises	□ laboratories
	☒ distance learning	☒ mentorship
	□ fieldwork	☒ other ____________________
6. Comments		

7. Student responsibilities

Students are required to apply the acquired knowledge and skills to complete work tasks, individually and in teams, in an actual working environment of the training provider. Students must continuously keep the internship journal (e.g. in the form of an e-portfolio).

8. Monitoring[50] of student work

Class attendance		Class participation		Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment		Report		Practical work	3
Portfolio (internship journal)	1	Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

While students complete their work tasks, the following items will be assessed: quality of completed work tasks (precision, thoroughness, quantity and speed), ability to learn (understanding and acquiring new skills and ideas), reliability, conscientiousness, accuracy, presence at the work place, acceptance of work tasks, acceptance of instructions and feedback and engagement, ability to cooperate (efficient team work, contribution to group activities).

10. Mandatory literature (at the time of submission of study programme proposal)

11. Optional/additional literature (at the time of submission of the study programme proposal)

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

[50] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Marija Brkić Bakarić, PhD	
Name of the course	Desktop and Mobile Application Development	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for SD module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to prepare students for developing GUI applications and to introduce them to specific and cross-platform technologies for mobile application development.

2. Course enrolment requirements

A pass mark in Object-Oriented Programming.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students will be able to:

I1. Develop an application supported by version control and software maintenance systems.

I2. Choose and recommend appropriate technologies for application development according to user requests.

I3. Develop and test a mobile application for a platform of your choice.

I4. Prepare a developed mobile application for distribution in an app store.

I5. Design and develop a cross-platform application according to particular user requests.

4. Course content

Development of GUI applications. Mobile platforms. Mobile application development technologies. Application lifecycle. MVC and MVVM architecture. Overview of application development components. Development of user interface for multiple screen resolutions and orientations. Advanced concepts (e.g. use of contacts, calendar and camera, maps and location services, use of push notifications). Testing and continuous integration. Store.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	E-learning system will be used in this course. When they enrol into the course, students will be instructed to use the tools available in the system. A detailed schedule will be provided in the syllabus.	

7. Student responsibilities

Students should regularly participate in all course activities (preliminary exams, projects, problem-solving tasks, homework) and score at least 50% on the final exam. A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[51] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam	0.5	Essay		Research	
Project	1.5	Continuous assessment	1.5	Report		Practical work	
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

For homework, ask students to finish a given code, with the support of a given software versioning system (I1).

In the written or oral exam (theoretical preliminary exam), test the understanding of various technologies for the development of desktop and mobile applications and their components (I2), e.g. choose a sentence that best describes the relationship between a delegate and an event.

In the written or oral exam (theoretical preliminary exam), ask students to determine critical elements of a given application development platform according to specific user requests (I2), e.g. pair a platform with a corresponding statement.

For homework or project, ask students to define the criteria that will be important for choosing an application development technology with specific functionalities (I2), e.g. according to the given user requests, choose and explain the choice of application development technology.

Through group or individual practical project assignment, ask students to create an application according to specific instructions and evaluation criteria (I3, I4), e.g. devise and create a simple application sending occasional motivational messages to the user.

In the written assessment (practical preliminary exam), ask students to create a cross-platform application according to specific user requests (I5), e.g. create a simple application for currency exchange.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Scripts, presentations and other learning material available in the e-course.
2. Jeff Johnson: Designing with the mind in mind, Elsevier, 2010.
3. Harvey Deitel, Paul Deitel: C# 6 for Programmers, Pearson Education, 2017.
4. https://developer.apple.com/documentation
5. https://docs.microsoft.com/hr-hr/windows

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Barbara Doyle: C# Programming: From Problem Analysis to Program Design, Course Technology, 2008.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

[51] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences		
Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).		

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Ana Meštrović, PhD	
Name of the course	Declarative Programming Languages	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for SD module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about the declarative programming paradigm. The objective is to teach students how to implement software solutions in functional and logic programming languages.

2. Course enrolment requirements

A pass grade in Programming 1 and Programming 2.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain basic properties and differences between declarative and imperative programming paradigms, as well as basic properties of functional and logic programming languages.

I2. Explain the theoretical basis of functional languages (Church-Turing thesis and lambda calculus) and logic languages (predicate calculus, Horn clause, definite clause, unification, resolution rule, backtracking).

I3. Apply elements of functional languages (referential opacity, higher-order functions, lazy evaluation, type inference) when completing simple problem-solving tasks.

I4. Implement software solutions for the given simple problem-solving tasks by using a specific functional language and by applying lists, list comprehension and recursive programming style.

I5. Model higher-order functions and function forms (map, fold and filter) and apply them in the implementation for various problem-solving tasks.

I6. Implement software solutions for the given simple problem-solving tasks by using a specific logic language and by defining facts, rules, recursive rules (implementation of a knowledge base).

I7. Apply skills and knowledge from declarative programming style in the implementation of a software solution for a given complex problem-solving task, by applying appropriate concepts of functional/logic programming and ready-to-use libraries in a given functional/logic language.

4. Course content

- Declarative programming paradigm. Comparison of basic programming paradigms.
- Theoretical basis of functional programming languages (Church-Turing thesis and lambda calculus) and logic programming languages (predicate calculus, Horn clause, definite clause, resolution rule).

- Motivation and fields of application of functional and logic programming styles. Overview of functional and logic programming languages.
- Introduction to functional programming. Basic properties of functional programming: referential opacity, higher-order functions, lazy evaluation, type inference.
- Lists, list comprehension, recursion.
- Higher-order functions. Function forms: map, filter, fold.
- Introduction to logic programming: knowledge representation (facts, rules, goals). Syntax, semantics (procedural and declarative). Finding answers to queries, unification process, resolution.
- Definition of recursive rules. Built-in predicates. Arithmetic and logic operations. Data structures: lists, trees. Cut. Defining a knowledge base.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system, prepare for classes (do homework) and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical quizzes and preliminary exams) and successfully pass them.
- Complete individual or team practical work related to a given topic.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[52] of student work

Class attendance	1	Class participation	0.5	Seminar paper	0.5	Experimental work	
Written exam		Oral exam		Essay		Research	

[52] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Project		Continuous assessment	1.5	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of declarative programming paradigm and theoretical postulates of functional and logic programming languages (I1, I2). For example, multiple choice questions, fill in the blank questions and/or essay questions in which students need to explain the relationship between declarative programming paradigm and lambda calculus, compare declarative and imperative programming paradigm.
- A seminar paper defined by the instructions and evaluation criteria set in advance in which students need to examine and describe properties of a programming language of their choice from the field of declarative programming paradigm (I1, I2). For example, examine and describe the properties of the programming language Logo.
- Practical assessment (problem-solving tasks, preliminary exam) in which students need to apply functional programming style to solve simple problem-solving tasks (I3, I4, I5). For example, implement a function for simple data encryption (e.g. Caesar cipher) by applying the map function form.
- Practical assessment (problem-solving tasks, preliminary exam) in which students need to apply logic programming style to solve simple problem-solving tasks (I6). For example, implement rules describing relations and rules from the knowledge base for projects and project managers.
- Group or individual practical project assignment in which students apply knowledge and skills in declarative programming style to solve a practical problem-solving task according to specific instructions and evaluation criteria (I3, I4, I5, I6, I7). For example, implement a module for processing character strings and preparing texts for statistical analysis.

10. Mandatory literature (at the time of submission of study programme proposal)

1. Meštrović, A. (2018). Online skripta: Deklarativni programski jezici – teorija i zadaci, Odjel za informatiku, Sveučilište u Rijeci, Rijeka 2018.
2. Thompson, S. (2011). Haskell: the craft of functional programming (Vol. 2). Addison-Wesley.
3. Lloyd, J. W. (2012). Foundations of logic programming. Springer Science & Business Media.
4. Sebesta, R. W. (2016). Concepts of programming languages.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Gallagher, J. P., & Sulzmann, M. (2018). Functional and Logic Programming. Springer.
2. Bird, R. (1998). Introduction to functional programming using Haskell (Vol. 2). Hemel Hempstead, UK: Prentice Hall Europe.
3. Sterling, L., & Shapiro, E. Y. (1994). The art of Prolog: advanced programming techniques. MIT press.
4. Révész, G. E. (2009). Lambda-calculus, combinators and functional programming. Cambridge University Press.
5. Brodie, M. L., Mylopoulos, J., & Schmidt, J. W. (Eds.). (2012). On conceptual modelling: Perspectives from artificial intelligence, databases, and programming languages. Springer Science & Business Media.
6. Petricek, T., & Skeet, J. (2009). Real World Functional Programming: With Examples in F# and C.

Manning Publications Co.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Asst. Prof. Miran Pobar, PhD	
Name of the course	Introduction to Embedded Systems and the Internet of Things	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for CS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about embedded computer systems, their application and technologies and to develop skills necessary to plan and develop practical projects.

2. Course enrolment requirements

A pass mark in *Computer Architecture and Organization*.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe typical properties of embedded computer systems and networked devices.

I2. Analyse trends in the technological development of embedded computer systems and their effect on fields of application.

I3. Analyse and compare possibilities and limitations of wireless and mobile network technologies for the Internet of Things.

I4. Choose appropriate peripherals of the embedded system depending on the purpose, users, timing constraints, available budget and technologies.

I5. Analyse the behaviour of the embedded computer system by using a simulation tool.

I6. Create appropriate project documentation (including goals, user analysis, functional description and model).

I7. Create an embedded computer system that includes sensors, actuators and software by using appropriate hardware and development tools based on a previously created project plan.

4. Course content

- Properties of embedded systems, typical characteristics and fields of application
- Embedded systems and the Internet of Things (IoT)
- Hardware and software platforms for the development of embedded systems
- Sensors, drivers
- IoT standards
- Communication and network technologies and standards for the Internet of Things
- Example of an embedded system – case study

- Modelling and simulation of an embedded computer system

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Prepare an individual or group project on a given topic, in a written form, and present it to lecturers and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[53] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	0.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts related to embedded computer systems (I1, I2, I3). For example, students through multiple choice questions, fill in the blank questions and essay questions list the characteristics of modern embedded computer systems, choose an appropriate communication technology for a given problem (e.g. wearables, in-car devices) and provide valid reasons for their choice, describe the application of embedded systems in certain fields and

[53] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

comment on their potential development (e.g. in sports, home automation, agriculture).

- Practical assignments (assignments during exercises to obtain credits) in which students simulate the behaviour of a certain segment in the embedded system by applying tools and software for the simulation of embedded computer systems, e.g. choosing an appropriate accelerometer to be used with Arguino platform and simulating its use (I4, I5).
- Group or individual work in the form of preparing appropriate project documentation for a larger practical project on a topic of their choice, which includes goals, user analysis and functional description and model of the system. Students will receive the instructions and evaluation criteria for the seminar in advance (I4, I5, I6).
- Group or individual seminar paper in the form of practical work, performed by using an appropriate hardware and software platform and peripherals, based on previously prepared project documentation for the chosen topic of the practical project. Students will receive the instructions and evaluation criteria for the seminar in advance (I7).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Marwedel P. Embedded system design, 3rd ed. New York: Springer; 2017.
2. Edward A. Lee and Sanjit A. Seshia, *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, Second Edition, MIT Press, ISBN 978-0-262-53381-2, 2017.
3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Claire Rowland, Elizabeth Goodman, Martin Chalier, Ann Light, Alfred Lui, *Designing Connected Products: UX for the Consumer Internet of Things*, 2015, O’Reilly Media, Inc, ISBN 978-1449372569
2. White, Elecia. *Making Embedded Systems: Design Patterns for Great Software*. " O'Reilly Media, Inc.", 2011.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Ivo Ipšić, PhD / Vedran Miletić, PhD	
Name of the course	Computer System Administration	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for CS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about computer system administration and develop skills for using operation system installation and upgrade tools, as well as monitoring and administration tools.

2. Course enrolment requirements

A pass mark in Operating Systems and Computer Networks.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Write internal documentation for a computer system.
I2. Prepare a virtual machine with an operating system, installed software and settings of computer resources, networking and data storage according to specific instructions.
I3. Manage the existing operating system services and create own services.
I4. Manage users and groups in the operating system, especially to control file access.
I5. Connect multiple real or virtual computers in a network with specified properties and limitations.
I6. Manage operating system start-up process and kernel settings.
I7. Build an operating system kernel, drivers and application software.

4. Course content

- Basic concepts of computer system administration. Use of existing documentation. Creation of internal documentation.
- Virtualization. Full virtualization. Container-based virtualization.
- Operating system installation. Software installation. User and group management. Permissions and access control.
- Data storage management. Data backup.
- Service management. Task automation. Event tracking.
- Computer system start-up process. Management of operating system kernel. Hardware drivers.
- Restricting local access. Restricting network access.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☐ multimedia and network
	☒ exercises	☒ laboratories

	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work and computer laboratory work, with the application of a learning management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Complete individual or team practical work related to a given topic.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[54] of student work

Class attendance	1	Class participation		Seminar paper		Experimental work	
Written exam		Oral exam	1	Essay		Research	
Project		Continuous assessment	2	Report		Practical work	1
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Practical assessment on a computer (practical preliminary exam) in which students, with the help of existing documentation, demonstrate their ability to use computer system administration tools, e.g. for managing users, groups, permissions, services, data storage, kernel and operating system start-up process, and software development tools (I2, I3, I4, I6, I7).
- Group or individual practical work in which students, according to specific instructions, set up a virtual machine (or machines) with a specific operating system, applications and services, and write internal documentation for installing and adjusting the machine (or machines) (I1, I2, I3, I4, I5).
- Oral assessment in which students demonstrate their understanding of theoretical concepts related to computer system administration, for example through essay questions and demonstration of skills (I3, I4, I5, I6, I7).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Frisch, Ae. Essential system administration. (O'Reilly, 2002).
2. Nemeth, E., Snyder, G., Hein, T. R., Whaley, B. & Mackin, D. Unix and Linux system administration handbook. (Addison-Wesley, 2017).

[54] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

3. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Hennessy, J. L. & Patterson, D. A. Computer architecture: a quantitative approach. (Morgan Kaufmann, 2007).
2. Limoncelli, T., Hogan, C. J. & Chalup, S. R. The practice of system and network administration. (Addison-Wesley, 2007).
3. Beyer, B., Jones, C., Petoff, J. & Murphy, N. R. Site reliability engineering: how Google runs production systems. (O'Reilly, 2016).
4. Doleželová, M., Muehlfeld, M., Svistunov, M., Wadeley, S., Čapek, T., Hradílek, J., Silas, D., Heves, J., Kovář, P., Ondrejka, P., Bokoč, P., Prpič, M., Slobodová, E., Kopalová, E., Svoboda, M., O'Brien, D., Hideo, M., Domingo, D. & Ha, J. System administrator's guide. (Red Hat, 2018).
5. Aoki, O. Debian reference. (Debian, 2018).
6. The FreeBSD documentation project. FreeBSD handbook. (FreeBSD, 2018.)

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Marina Ivašić-Kos, PhD	
Name of the course	Fundamentals of Game Development	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for MMS module	
Year of study	2	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic concepts related to the types of computer games, design of computer games and development of computer games. Students will be taught how to work out and independently design a computer game, determining its content, goal and rules, and how to develop its prototype using a development environment for creating computer games and interactive 2D and 3D contents, all by applying basic principles of design and practical knowledge about game development.

2. Course enrolment requirements

Previously taken course Object-Oriented Programming.

3. Expected learning outcomes

After successfully fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe basic concepts of game design and analyse computer game design and formal elements of a game.

I2. Compare game prototypes of different genres and compare basic elements and algorithms used in typical genres.

I3. Build basic elements of a simple game world using a development environment for creating computer games and interactive 2D and 3D contents, using built-in primitive forms and tools (creating visual objects, controlling input devices, creating animation, graphical user interface, saving and loading data, working with sound).

I4. Create and modify different computer game mechanics (object movement, collision detection, interaction with objects, shooting, random generation of objects, level generation).

I5. Apply appropriate artificial intelligence algorithms and integrate them into the game or use scripts in an appropriate programming language (e.g. C#) to establish main functionality of the game;

I6. Apply theoretical basis related to design and development of a computer game when creating a prototype of own computer game.

4. Course content

- Introduction to the development of computer games. Overview of game development through history.
- Types and genres of computer games (platform games, action games, adventure games, logic games, word games, casual games, educational games, first-person games). Basics of game design

and formal elements of a game. Game prototypes of different genres and general guidelines for video game development (story, goal, rules, playability).

- Basics of using a development environment (e.g. Unity). Creation of a simple game: creating and rearranging primitive shapes, working with cameras.
- Graphics: character appearance, perspective, texture, colour. Creation of new materials and their application to models.
- Physics and object movement. Mechanics in computer games: detection of collision between objects, interaction with objects. Shooting. Random generation of objects.
- Introduction to the internal structure of game code based on concepts of object-oriented programming. Object-oriented programming with a focus on C# – classes and objects, interfaces, data access, data processing.
- Animation. 2D and 3D game models. Controlling 3D animated characters.
- Elements of graphical user interface for manipulating objects in 2D and 3D space.
- Controlling input devices (mouse, keyboard). Game control, accurate object control and receiving feedback. Saving and loading data.
- Working with sound. Adding audio effects and music.
- Artificial intelligence algorithms in games (pathfinding, perception, reasoning).
- Increasing the game complexity, generating levels.
- Examples and analyses of design and implementation solutions for computer games of different genres.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams).
- Design, create and present a prototype of a computer game, and score at least 50% on the final exam.

A detailed scoring system for the course will be specified in the course syllabus.

8. Monitoring[55] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students describe basic concepts of game design and propose concepts for the development of a simple computer game (I1, I2).
- Practical assessment (preliminary exam on a computer) in which students add basic elements of a computer game to the game world using a game development tool and implement an algorithm for a specific type of mechanics, as well as certain computer game functionalities (I3, I4).
- Practical assessment (preliminary exam on a computer) in which students develop a prototype for a game of a specific genre using a game development tool, test the game, modify algorithms and functionalities that do not fulfil the needs and embed new functionalities (I3, I4).
- Practical project assignment in which students apply the theoretical basis and knowledge about the development of computer games, and work out and create a computer game according to the instructions and evaluation criteria set in advance (I5, I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. J. Gibson: Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C# Addison-Wesley, 2015.
2. J. Hocking: Unity in Action: Multiplatform Game Development in C# with Unity 5 Shelter Island NY: Manning Publications, 2015.
3. LCF Publishing: C#: Learn C# in One Day and Learn It Well.
4. I. Millington: Artificial Intelligence for Games (The Morgan Kaufmann Series in Interactive 3D Technology) 1st Edition, 2006.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. S. Rogers: Level Up!: The Guide to Great Video Game Design John Wiley & Sons, 2010.
2. D. Schuller: C# Game Programming: For Serious Game Creation, Cengage Learning PTR; 1st edition, 2010.
3. S. Blackman: Beginning 3D Game Development with Unity, Apress, 2011.
4. T. Miller; "Beginning 3D Game Programming", Sams Publishing, 2004.
5. J. Albahari: C# 7.0 in a Nutshell: The Definitive Reference.

12. Number of assigned reading copies in relation to the number of students currently attending the course

[55] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Nataša Hoić-Božić, PhD	
Name of the course	Multimedia Design	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for MMS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge and develop skills necessary to plan and create multimedia projects. Students will be introduced to planning strategies, creation methods and techniques and appropriate software for producing multimedia presentations according to already prepared project specifications.

2. Course enrolment requirements

A pass mark in Multimedia Systems.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Analyse development trends in multimedia technologies and their effect on the design and development of multimedia content.

I2. Design and organise information in order to present them to users in multimedia formats in an efficient and visually attractive manner.

I3. Determine the suitable type of multimedia presentation depending on the purpose, users, timing constraints, available budget and technologies.

I4. Apply techniques and methods of information technology project management to multimedia projects and prepare the necessary project documentation (including goals, user analysis, scenarios with interface diagrams and navigation).

I5. Create a multimedia presentation (with attractive visual design and containing images, audio, video, animation, hypertext and interactivity) using an appropriate programming tool, based on a previously made project plan and in accordance with multimedia standards.

I6. Design and create a professional digital multimedia portfolio that includes multimedia contents created during the course.

4. Course content

- Current development of multimedia and individual media elements (hypertext, 2D and 3D graphics and animation, video, audio) and modern development trends in multimedia technologies (augmented reality, virtual reality...).
- Accepted design standards and rules for users of visually attractive media elements and presentations.

- Types of multimedia presentations and applications, and the overview of programming tools for their production.
- Project approach to planning, managing and creating multimedia contents.
- Professional digital multimedia portfolio.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical preliminary exams) and successfully pass them.
- Create and continuously update a digital portfolio (e-portfolio) consisting of contents created during the course.
- Write an individual or group paper on a given topic and present it to lecturers and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[56] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio	0.5	Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of theoretical concepts regarding multimedia technologies (I1, I2, I3). For example,

[56] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

students through multiple choice questions, fill in the blank questions and essay questions name the characteristics of modern multimedia technologies (e.g. augmented reality), describe a specific type of multimedia presentation (e.g. computer game, 3D animation, web), explain why a shown multimedia presentation is well designed or poorly designed.

- Group or individual seminar paper in the form of prepared project documentation for a specific type of large multimedia project (computer game, 2D/3D multimedia presentation, multimedia website...), including goals, user analysis, scenarios with interface diagrams and navigation. Students will receive the instructions and evaluation criteria for the seminar in advance (I4).
- Group or individual seminar paper in the form of a multimedia presentation made using an appropriate programming tool (e.g. Blender, Unity) based on previously prepared project documentation for a specific type of large multimedia project. Students will receive the instructions and evaluation criteria for the seminar in advance (I5).
- E-portfolio, or digital multimedia portfolio, consisting of multimedia contents created during the course and student's learning diary. Completeness and quality of contents in the portfolio will be evaluated according to the criteria set in advance (I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Costello, V. (2016.) Multimedia Foundations: Core Concepts for Digital Design 2nd Edition.
2. Hoić-Božić, N. (2015). Multimedijski sustavi, Online skripta s predavanjima u Moodle e-kolegiju.
3. Hoić-Božić, N. (2018). Uvod u web dizajn, Online skripta s predavanjima u Moodle e-kolegiju.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Vaughan, T. (2014). Multimedia: Making It Work, Ninth Edition 9th Edition, Berkeley: McGraw-Hill Osborne Media.
2. Osborn, T. (2018). Hello Web Design: Design Fundamentals and Shortcuts for Non-Designers.
3. Niederst Robbins, J. (2018). Learning Web Design, 5th Edition (A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics), O'Reilly Media, http://www.learningwebdesign.com/
4. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Sanda Martinčić-Ipšić, PhD / Lucia Načinović Prskalo, PhD	
Name of the course	Introduction to Data Analysis and Visualization	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for IS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire basic knowledge about the processes of data analysis and visualization, to introduce students to the processes of data collection and adjustment, to teach them how to perform various organization and management procedures on data sets, apply basic statistical methods for the purpose of data analysis, basic graphical methods of exploratory analysis and various methods of data visualization.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Describe basic concepts, stages and procedures in the process of data analysis and visualization.

I2. Apply methods of basic descriptive statistics to different data sets.

I3. Collect, adjust and clean data sets from various types of sources and of different file formats.

I4. Perform organization and management operations on data sets, such as choosing, joining, transforming, filtering, sorting, aggregation and grouping of data.

I5. Apply basic statistical methods to analyse a data set.

I6. Apply basic graphical methods of exploratory data analysis to different data sets.

I7. Choose and apply an appropriate type of chart or representation for specific data sets in order to better identify and understand the problem.

4. Course content

- Concepts of data, data analysis and data visualization, historical overview, application of data analysis and visualization.
- Working with different data structures – time series, data frames, indexed objects.
- Application of basic descriptive statistics to data sets – centrality measures, dispersion, correlation etc.
- Collection and cleaning of data from different types of sources, different data set file formats.
- Application of different organization and management operations on data sets – transformation, choosing, joining, filtering, sorting, grouping, aggregation of data etc.

- Basic statistical methods used in the process of data analysis (hypothesis testing, linear models, multiple factors, variance analysis etc.).
- Basic methods of exploratory data analysis.
- Data visualization – basic principles of good charts and representations.
- Application of different visualization methods and different chart and representation types to different data sets.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	☒ multimedia and network
	☒ exercises	☒ laboratories
	□ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	E-learning system will be used in this course. When they enrol into the course, students will be instructed to use the tools available in the system. A detailed schedule will be provided in the syllabus.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes and actively participate in the learning process.
- Follow course activities within the distance learning system.
- Complete given assignments and hand them in within the given deadline.
- Participate in continuous assessment.
- Complete an individual or group project assignment according to the instructions provided and present it to the lecturer and other students.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[57] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1.5	Continuous assessment	0.5	Report		Practical work	0.5
Portfolio							

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Completing and handing in the assignments within the given deadline – during this course students will be given short assignments to be completed in class or at home, for the purpose of testing their

[57] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

knowledge and understanding of basic concepts and procedures in the process of data analysis and visualization (I1), their ability to apply methods of descriptive statistics to data sets (I2), degree of competence in the process of collection, adjustment and cleaning of data sets (I3) and of performing organization and management operations on data sets (I4), their ability to apply basic statistical methods in data analysis (I5), basic graphical methods of exploratory analysis (I6) and various visualization methods to different data sets (I7). (For example, using a given tool or module of a programming language, visualize the life expectancy per continents for a specific year and choose the best representation method).

- Written or online assessment (preliminary exam) in which students demonstrate their understanding of basic concepts related to data analysis and application of different descriptive statistics processes to data sets (I1, I2). (For example, test the understanding of frequency histogram through multiple choice questions, fill in the blank questions and/or essay questions).
- Group or individual project assignment in which students need to collect a data set, perform various preparation, cleaning, adjustment, organization and management operations on it, apply chosen statistical methods and graphical methods of exploratory analysis and visualization according to specific instructions, and present the results of the analysis to the lecturer and other students (I2, I3, I4, I5, I6, I7).
- Written or online assessment (final exam) in which students demonstrate their understanding of concepts and procedures from the field of data analysis and visualization (I1, I2, I3, I4, I5, I6, I7) (For example, explain the application and purpose of a line graph on a given data set).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Grus, J.: Data Science from Scratch, First Principles with Python 1st Edition, O'Reilly Media, 2015.
2. Chen, C., Härdle, W., Unwin A.: Handbook of Data Visualization, Springer, 2008.
3. McKinney, W.: Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 2nd Edition, O'Reilly Media, 2017.
4. Embarak, O.: Data Analysis and Visualization Using Python, Apress, 2018.
5. VanderPlas, J.: Python Data Science Handbook, O'Reilly Media, 2017.
6. Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Introduction to Data Analysis Handbook, Migrant & Seasonal Head Start Technical Assistance Center, AED/TAC, 2006.
2. Downey, B. A.: Think Stats, Exploratory Data Analysis in Python, Green Tea Press, Needham, 2014.
3. Stanton, J.: An Introduction to Data Science, Syracuse University, 2013.
4. Madhavan, S. Mastering Python for Data Science, Packt Publishing, 2015.
5. Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Dana, John Wiley & Sons, Inc, Indianapolis, 2015.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences		
Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).		

COURSE DESCRIPTION		
Course instructor	Full Prof. Patrizia Poščić, PhD / Danijela Jakšić, PhD	
Name of the course	Next Generation Databases	
Study programme	Undergraduate university programme in informatics	
Status of the course	**compulsory** for IS module	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to master expanded knowledge about databases, with a focus on data warehouses and NoSQL databases. Such knowledge includes, but is not limited to, conceptual and logic design (modelling) of data warehouses, OLAP, distinguishing between several types of NoSQL databases and following current development trends in database technologies.

2. Course enrolment requirements

A pass mark in Databases and Data Modelling.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain various approaches to data warehouse (DW) development.

I2. Compare properties of different types of DW architecture.

I3. Distinguish between different DW design (modelling) methods with respect to layers of chosen DW architecture.

I4. Create a complete DW model (relational, dimensional, Data Vault) for a chosen DW architecture.

I5. Explain techniques and approaches to business data analysis (OLAP analysis, business reporting, Big Data).

I6. Distinguish between different types and approaches to NoSQL DW design.

I7. Explain basic concepts of distributed DWs.

4. Course content

Decision support systems. Data warehouses. Data warehouse architecture. Dimensional model. Data Vault model. ETL processes. OLAP. Business Intelligence. DW 2.0. Big Data (high-volume data). NoSQL databases. Distributed databases. Development trends in database technologies.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☒ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held by combining classroom work, computer laboratory work and individual work outside the classroom, using a learning	

	management system (LMS). When they enrol into this course, students will be instructed to use the distance learning system. A detailed schedule with lectures and exercises will be defined in the syllabus.

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly attend classes, actively participate in all course activities and follow course activities within the distance learning system.
- Participate in continuous assessment (preliminary exams and quizzes) and achieve the number of credits equal to or higher than the passing score (if any).
- Complete a team project assignment (practical work) on a given topic, in writing, and individually score at least 50% of credits for it.
- Write a research seminar paper on a given topic and in required form and individually score at least 50% of credits for it.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[58] of student work

Class attendance	1	Class participation	0.5	Seminar paper	1	Experimental work	
Written exam		Oral exam		Essay		Research	0.5
Project	1	Continuous assessment	1	Report		Practical work	
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through activities in class, continuous assessment (preliminary exams and quizzes), a practical project assignment and research seminar paper, in addition to computer work.

- In the theoretical assessment, students demonstrate their understanding of theoretical concepts related to data warehouses (I1) and distributed DBs (I7), compare different types of DW architecture (I2) and types/approaches to building of NoSQL DBs (I6), and distinguish between different DW design (modelling) methods (I3), as well as between different techniques of and approaches to business data analysis (I5). For example, explain the difference between Kimball and Inmon approach to the building of DWs or explain the main difference between column-oriented and document-oriented NoSQL databases.
- In the practical preliminary exam, students work in a given NoSQL DB environment on a computer. E.g. the student independently creates objects and structures within a NoSQL database (object creation, data input, data search etc.) on a computer. (I6)
- In the team project assignment, students complete practical work in teams, thus demonstrating

[58] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

their understanding of DW design (modelling) methods (I4). This practical work includes the creation of a complete DW model for a chosen DW architecture. For example, creation of a set of models for banking business, including the relational/entity-relationship model of data sources, Data Vault model of central/integrated DW and dimensional model of data marts, as well as queries for data analysis.

- The seminar paper includes research on a given topic and in a given form, and the presentation of the research in writing (seminar - scientific article) (I1-I7). For example, writing a research seminar paper on the topic "Comparison of DW design methodologies”.

10. Mandatory literature (at the time of submission of study programme proposal)

1. M. Golfarelli, S. Rizzi (2009). Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill.
2. R. Kimball, M. Ross (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition. Wiley.
3. W.H. Inmon, D. Strauss, G. Neushloss (2008). DW 2.0: The Architecture for the Next Generation of Data Warehousing. Morgan Kaufman.
4. C. Adamson (2010). Star Schema: The Complete Reference. McGraw-Hill.
5. D. Linstedt, M. Olschimke (2015). Building a Scalable Data Warehouse with Data Vault 2.0. Morgan Kaufman.
6. D. Sullivan (2015). NoSQL for Mere Mortals. Addison-Wesley.
7. M. Tamer Özsu, P. Valduriez (2011). Principles of Distributed Database Systems. Springer.
8. D. Jakšić (2018). Scripts, presentations and other learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. W.H. Inmon (2005). Building the Data Warehouse. Wiley.
2. R. Kimball, J. Caserta (2004). The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley.
3. R. Kimball, M. Ross, W. Thorntwaite, J. Mundy, B. Becker (2008). The Data Warehouse Lifecycle Toolkit, 2nd Edition. Wiley.
4. A. Noguès, J. Valladares (2017). Business Intelligence Tools for Small Companies: A Guide to Free and Low-Cost Solutions. Apress.
5. Appropriate software manuals.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Assoc. Prof. Ana Meštrović, PhD	
Name of the course	Programming for Complex Problem Solving	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is for students to acquire knowledge about advanced programming techniques. The objective is to teach students how to solve complex problem-solving tasks.

2. Course enrolment requirements

There are no course enrolment requirements.

3. Expected learning outcomes

After fulfilling all the obligations anticipated by the course, students are expected to be able to:

I1. Explain the steps of more complex algorithms on trees and graphs.

I2. Explain hash procedures and solve collision problems.

I3. Analyse and compare different encryption algorithms.

I4. Apply and adjust appropriate existing algorithms and data structures to solve more complex similar/analogous problems.

I5. Develop computer programs with implemented algorithms for solving complex problems.

I6. Critically evaluate the quality of different software solutions for a given problem and choose the best solution according to specific criteria.

4. Course content

- Introductory examination of complex problems. Revision: algorithm, algorithm complexity.
- Introductory examination of complex problems and algorithms.
- More complex algorithms on trees and graphs.
- Hash function. Solving collision problems.
- Encryption algorithms.
- Procedures for solving complex mathematical problems.
- More complex algorithms for working with character strings.

5. Manner of instruction	☒ lectures	☒ individual assignments

	□ seminars and workshops	□ multimedia and network
	☒ exercises	☒ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments	Classes are held in blended form, by combining classroom work, individual work outside the classroom and e-learning, using a learning management system (LMS). A detailed schedule with online lessons and classroom lectures will be defined in the syllabus. When they enrol into this course, students will be instructed to use the tools available in the system.	

7. Student responsibilities

Student responsibilities for this course are as follows:

- Regularly follow course activities within the distance learning system, prepare for classes (do homework) and attend classes taking place in the form of lectures, auditory and/or laboratory exercises.
- Participate in continuous assessment (theoretical and practical quizzes and preliminary exams) and successfully pass them.
- Complete individual or team practical work related to a given topic.
- Score at least 50% on the final exam.

A detailed scoring system for the course and passing scores for individual activities will be specified in the course syllabus.

8. Monitoring[59] of student work

Class attendance	1	Class participation	0.5	Seminar paper	0.5	Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment	1.5	Report		Practical work	1.5
Portfolio		Discussion					

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

- Written or online assessment (theoretical preliminary exam) in which students demonstrate their understanding of concepts related to advanced algorithms from the field of computing (I1, I2, I3).
- Practical assessment (problem-solving tasks, preliminary exam) in which students need to identify algorithms and data structures and know how to adapt and apply them to solve a similar problem (I1, I3, I4).
- Practical assessment (problem-solving tasks, preliminary exam) in which students solve complex problem-solving tasks and assess solution complexity and quality (I5).

[59] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

- Practical project assignment in which students implement the solution to a complex practical problem-solving task according to specific instructions and evaluation criteria (I5, I6).

10. Mandatory literature (at the time of submission of study programme proposal)

1. Kusalić, D. (2010). Napredno programiranje i algoritmi u Cu i C++-u. Zagreb, Element.
2. Ahuja, R. K. (2017). Network flows: theory, algorithms, and applications. Pearson Education.
3. Skiena, S. S. (1998). The algorithm design manual: Text (Vol. 1). Springer Science & Business Media.
4. Scripts, presentations and learning material available in the e-course.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.
2. Levitin, A. (2012). Introduction to the design & analysis of algorithms. Boston: Pearson.
3. Kocay, W., & Kreher, D. L. (2016). Graphs, algorithms, and optimization. chapman and hall/cRc.
4. Ellis, B. (2014). Real-time analytics: Techniques to analyze and visualize streaming data. John Wiley & Sons.
5. Galbraith, S. D. (2012). Mathematics of public key cryptography. Cambridge University Press

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION		
Course instructor	Full Prof. Dean Crnković, PhD	
Name of the course	Discrete Mathematics	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The objective of the course is to introduce students to the graph theory and combinatorial thinking and combinatorial argument. For this purpose, during this course students need to:

1. define basic concepts of graph theory, and describe and analyse basic properties of graphs,
2. define graph connectivity, and analyse properties related to connectivity,
3. describe a problem and analyse shortest path algorithms (Dijkstra's algorithm, Floyd’s algorithm),
4. define Eulerian and Hamiltonian graphs and analyse their properties,
5. describe a connectivity problem and analyse algorithms for finding the optimal tree (Kruskal's algorithm, Prim's algorithm),
6. define concepts related to graph colouring, analyse the corresponding properties and colouring problems,
7. define planar graphs and analyse their properties,
8. analyse polyhedral graphs and describe their properties,
9. analyse and compare certain algorithms.

2. Course enrolment requirements

Previously taken course Combinatorics.

3. Expected learning outcomes

After taking the course and passing the exam, students will be able to:

I1. Distinguish between the said concepts and graph properties, and provide valid arguments for applying appropriate properties and statements in task solving.

I2. Analyse graph connectivity problems and related properties.

I3. Analyse and provide valid reasons for applying the appropriate procedure for finding the shortest path.

I4. Analyse Eulerian and Hamiltonian graphs and provide valid arguments for applying certain definitions and properties in task solving.

I5. Solve connectivity problems and apply the algorithm for finding the optimal tree.

I6. Analyse graph colouring problems, and provide valid arguments for applying the appropriate procedures in solving the said problems.

I7. Provide valid arguments for the use of planar graph properties in task solving.

I8. Analyse polyhedral graphs and describe their properties.
I9. Compare certain algorithms and apply them to solve problems.
I10. Mathematically prove the foundation of procedures and formulas used within this course.

4. Course content

Introduction. Concept and basic properties of graphs. Incidence and adjacency matrices. Degree of a vertex. Walks, paths, cycles. Shortest path problem. Trees. Connectivity problem. Eulerian tours and Hamiltonian cycles. Weighted graph. Travelling salesman problem. Chinese postman problem. Graph connectivity. Reliable communication networks. Graph colouring. Brooks' theorem and Vizing's theorem. Chromatic polynomial. Planar graphs. Euler's formula. Polyhedral graphs.

5. Manner of instruction	☒ lectures	☒ individual assignments
	☐ seminars and workshops	☒ multimedia and network
	☒ exercises	☐ laboratories
	☒ distance learning	☐ mentorship
	☐ fieldwork	☐ other ____________________

6. Comments	The program of the course Discrete Mathematics is in correlation with other mathematical courses, especially with Combinatorics, and with informatics courses such as Algorithms and Data Structures and Introduction to Theoretical Computing.

7. Student responsibilities

Students must attend the classes, actively participate in all forms of classes, achieve a certain number of credits during the semester and pass the final exam (details will be provided in the course syllabus).

8. Monitoring[60] of student work

Class attendance	1	Class participation	0.5	Seminar paper		Experimental work	
Written exam	1.8	Oral exam	1.1	Essay		Research	
Project		Continuous assessment	0.6	Report		Practical work	
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through short tests, preliminary exams and the final exam.

- In the written assessment (tests), students demonstrate their understanding of theoretical concepts from the course material by applying their knowledge to specific tasks and by analysing properties (I1, I2, I3, I4, I5, I6, I7, I8, I9). For example, determine a degree sequence of a graph, whether the given sequence is graphical, determine adjacency matrix and incidence matrix of a given graph, determine girth and circumference of a given graph etc.
- In the written assessment (preliminary exams), students demonstrate that they have mastered the course material through mathematical modelling of a problem, application of knowledge to specific

[60] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

assignments, analysis of properties and choice of methods for solving the problem (I1, I2, I3, I4, I5, I6, I7, I8, I9). For example, analyse whether the given graphs are isomorphic by using their properties, determine whether the graphs are Eulerian, whether they are Hamiltonian, determine the optimal tree in the graph by using appropriate algorithms, as well as the shortest path between given points on the graph etc.

- In the written and oral assessment (final exam), students demonstrate that they have mastered the theoretical concepts of the course material through mathematical modelling of a problem, formulation of mathematical statements, expressing of theorems, analysis of properties and discussion of examples and proving them with an appropriate method (I1, I2, I3, I4, I5, I6, I7, I8, I9, I10). For example, define the concept of Eulerian graph, express and explain the theorem of characterization of Eulerian graphs using vertex degrees, and determine which graphs are Eulerian on the examples and why.

10. Mandatory literature (at the time of submission of study programme proposal)

1. D. Veljan: Kombinatorika i diskretna matematika, Algoritam, Zagreb, 2001.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. N. Biggs: Discrete Mathematics, Clarendon Press, Oxford, 1989.
2. R. Diestel: Graph Theory, Second edition, Springer-Verlag, New York, 2000.
3. R. Balakrishnan, K. Ranganathan: A Textbook of Graph Theory, Springer-Verlag, Heidelberg, 2000.
4. R. Balakrishnan: Schaum's outline of Graph Theory: Included Hundreds of Solved Problems, McGraw-Hill, New York, 1997.
5. C. L. Liu: Elements of Discrete Mathematics, McGraw-Hill, New York, 1987.
6. L. Lovasz: Combinatorial Problems and Exercises, North-Holland, Amsterdam, 1979.
7. F. Robert: Applied Combinatorics, Prentice Hall, Englewood Cliffs, 1984.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students' achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

COURSE DESCRIPTION

Course instructor	Asst. Prof. Tajana Ban Kirigin, PhD	
Name of the course	Set Theory	
Study programme	Undergraduate university programme in informatics	
Status of the course	**elective**	
Year of study	3	
ECTS credits and manner of instruction	ECTS credits	5
	Number of class hours (L+E+S)	30+30+0

1. Course objectives

The main objective of the course is to introduce students to basic concepts and results of the set theory and to teach students how to apply them. For this purpose, during this course students need to:

- define basic concepts of intuitive set theory,
- describe basic understanding of infinity in the set theory,
- define basic concepts and principles of cardinal arithmetic,
- define basic understanding of ordinal numbers and principles of ordinal arithmetic,
- introduce the issue of intuitive set theory (paradoxes).

2. Course enrolment requirements

Previously taken course Mathematical Logic.

3. Expected learning outcomes

After taking the course and passing the exam, students will be able to:

I1. describe the concept of set and present the issue of intuitive set theory, i.e. the need to axiomatically develop the set theory (A6, B6, E5, F6)
I2. describe basic set operations and generalize union and intersection operations on arbitrary families of sets (A7, B7, D5, E5, F6)
I3. formulate the concept of relation and function, and analyse and classify them according to their properties (A7, B7, D5, E5, F6)
I4. formulate the concepts of finite and infinite sets, enumerable and non-enumerable sets and classify them according to valid arguments (A6, B6, D5, E5, F6)
I5. formulate the concept of cardinal number and solve tasks and apply theorems from the field of arithmetic and cardinal number order with valid arguments (A6, B6, D5, E5, F6)
I6. formulate the concept and analyse the properties of similarity and order characteristics of sets (A6, B6, D5, E5, F6)
I7. formulate the concept of ordinal number and distinguish between and compare ordinal numbers with valid arguments (A6, B6, D5, E5, F6)
I8. solve tasks and apply theorems from the field of arithmetic and ordinal number order with valid arguments (A6, B6, D5, E5, F6)
I9. mathematically prove the foundation of all procedures and formulas used within this course (A6, B6, D5, E5, F6).

4. Course content

Introduction. Intuitive concept of set. Sets and classes. Algebra of sets. Arbitrary unions and intersections. Relative complement and DeMorgan's laws. Power set. Cartesian product of sets. Relations. Binary relations. Equivalence relation. Order relation. Good order. Principle of transfinite induction. Functions. Equipotent sets. Finite and infinite sets. Enumerability and non-enumerability. Cardinal numbers. Order and arithmetic of cardinal numbers. Similarities and order type. Similarities on well-ordered sets. Ordinal numbers. Order and arithmetic of ordinal numbers. Paradoxes of set theory and need for axiomatization of set theory. First-order theories: syntax, semantics. Prenex normal forms. Main test for first-order logic. Basic meta-results and limitations of first-order logic.

5. Manner of instruction	☒ lectures	☒ individual assignments
	□ seminars and workshops	☒ multimedia and network
	☒ exercises	□ laboratories
	☒ distance learning	□ mentorship
	□ fieldwork	□ other ____________________
6. Comments		

7. Student responsibilities

Students must attend the classes, actively participate in all forms of classes, achieve a certain number of credits during the semester and pass the final exam (details will be provided in the course syllabus).

8. Monitoring[61] of student work

Class attendance	1	Class participation		Seminar paper		Experimental work	
Written exam	1.5	Oral exam	1.5	Essay		Research	
Project		Continuous assessment	1	Report		Practical work	
Portfolio		Discussion		Individual assignments			

9. Assessment of learning outcomes in class and at the final exam (procedure and examples)

The set of learning outcomes is assessed through short tests, preliminary exams and an oral exam.

- In the written assessment (tests), students demonstrate their understanding of theoretical concepts from the course material (I2, I3, I4, I5, I6, I7, I8, I9), for example through multiple choice questions and essay questions. For example, determine whether there is injection and/or surjection between the given sets.
- In the written assessment (preliminary exam), students demonstrate that they have mastered the course material by applying their knowledge to specific tasks, analysing and solving a given problem, as well as known relevant theoretical results (I2, I3, I4, I5, I6, I7, I8, I9). For example, determine whether a given set is enumerable.
- In the written and oral assessment (final exam), students demonstrate that they have mastered the theoretical concepts from the course material by defining certain concepts, discussing the examples, expressing and proving relevant properties and results (I1, I2, I3, I4, I5, I6, I7, I8, I9). For

[61] IMPORTANT: Enter the appropriate proportion of ECTS credits for each activity so that the total number of credits equals the ECTS value of the course. Use empty fields for additional activities.

example, express and prove Cantor’s theorem.

10. Mandatory literature (at the time of submission of study programme proposal)

1. P. Papić: Uvod u teoriju skupova, HMD, Zagreb, 2000.
2. S. Lipschutz: Set Theory and Related Topics, McGraw Hill, New York, 1964.

11. Optional/additional literature (at the time of submission of the study programme proposal)

1. M. Vuković – Teorija skupova, skripta PMF, Zagreb, 2013.
2. H. B. Enderton: Elements of Set Theory, Academic press, New York, 1977.
3. A. Levy: Basic Set Theory, Springer 1979.

12. Number of assigned reading copies in relation to the number of students currently attending the course

Title	Number of copies	Number of students

13. Quality monitoring methods that ensure the acquisition of exit knowledge, skills and competences

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme (as part of the activities of the Quality Assurance Committee at the Department of Informatics). In the last week of classes, students will evaluate the quality of classes using an anonymous questionnaire. Students’ achievements in the course will also be analysed (percentage of students who passed the course and their average grade).

3.3. STRUCTURE AND WORKFLOW OF THE STUDY PROGRAMME AND STUDENT RESPONSIBILITIES

In the first year of study, students enrol into courses defined by the study programme. In the following years of study, students enrol into courses based on the number of ECTS credits achieved in the previous academic year and on the enrolment requirements.

Courses in the first four semesters are compulsory for all students. When enrolling into the third year, students have to choose between one of the 4 modules offered:

1. Software Development (SD)
2. Communication Systems (CS)
3. Multimedia Systems (MMS)
4. Information Systems (IS).

In the third year of study, compulsory courses for each module account for 25 ECTS credits. In addition to these compulsory courses, students enrol into elective courses as defined by the study programme. Students achieve ECTS credits for elective courses by choosing between courses that are compulsory in other modules or from the group of elective courses for all third-year students. Students can choose one elective course (with minimum 3 ECTS credits) from the group of courses offered as elective courses by other constituents of the University of Rijeka in the current year.

In the sixth semester, students have to complete training as part of the course called Undergraduate Internship, in accordance with the Ordinance on Internship at Undergraduate Study Programmes.

Study is completed by defending the bachelor’s thesis. Students are entitled to defend their bachelor’s thesis after passing all the exams and fulfilling all the responsibilities set out in the study programme.

The workflow of the study programme is defined by the Ordinance on Study Programmes at the University of Rijeka, as well as general responsibilities, while specific student responsibilities are defined in the description of each course and corresponding syllabus published each year before the beginning of a semester.

1ST YEAR OF STUDY

	1st semester class hours/week			2nd semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Mathematics 1	2	2					60	6
Programming 1	2	2					60	6
Fundamentals of Informatics	2	2					60	5
Multimedia Systems	2	2					60	5
Fundamentals of Economics for IT students	2	2					60	5
English Language for IT Profession	2	2					60	4
Physical Education 1								
Mathematics 2				2	2		60	5
Programming 2				2	2		60	5
Operating Systems				2	2		60	5
Data Modelling				2	2		60	5
Computer Architecture and Organization				2	2		60	5
Fundamentals of Probability and Statistics				2	2		60	4
Physical Education 2								
Total ECTS credits	**12**	**12**	**-**	**12**	**12**	**-**		**31+29**

2ND YEAR OF STUDY

	3rd semester class hours/week			4th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Mathematics 3	2	2					60	5
Analysis of Business Processes	2	2					60	5
Object-Oriented Programming	2	2					60	5
Computer Networks	2	2					60	5
Databases	2	2					60	5
Introduction to Software Engineering	2	2					60	5
Operations Research				2	2	-	60	5
Introduction to Web Programming				2	2	-	60	5
Algorithms and Data Structures				2	2	-	60	5
Information Technology Project Management				2	2	-	60	5
Information Systems Development				2	2	-	60	5
Security of Information and Communication Systems				2	2	-	60	5
Total ECTS credits								**30+30**

3RD YEAR OF STUDY, MODULE Software Development – SD

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Programming Paradigms and Languages	2	2					60	5
Code Optimization	2	2					60	5
Web Programming	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Desktop and Mobile Application Development				2	2		60	5
Declarative Programming Languages				2	2		60	5
Elective course				2	2		60	5
Elective course (common elective course offered by the University of Rijeka/course from another module/common elective course)							60	3
Undergraduate Internship							60	4
Bachelor’s Thesis							60	8
Total ECTS credits								**30+30**

ELECTIVE COURSES FOR THE MODULE Software Development – SD

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Communication Networks	2	2					60	5
Network and Mobile Operating Systems	2	2					60	5
Social Network Analysis	2	2					60	5
Multimedia Technologies	2	2					60	5
Computer Graphics	2	2					60	5
Computer Animation	2	2					60	5

Database Administration and Security	2	2					60	5
User Interface and Interaction Design	2	2					60	5
Specific Purpose Information Systems	2	2					60	5
System Dynamics	2	2					60	5
Programming for Data Science	2	2					60	5
Introduction to Theoretical Computing	2	2					60	5
Combinatorics	2	2					60	5
Numerical Mathematics	2	2					60	5
Mathematical Logic	2	2					60	5
Introduction to Embedded Systems and the Internet of Things				2	2		60	5
Computer System Administration				2	2		60	5
Fundamentals of Game Development				2	2		60	5
Multimedia Design				2	2		60	5
Introduction to Data Analysis and Visualization				2	2		60	5
Next Generation Databases				2	2		60	5
Programming for Complex Problem Solving				2	2		60	5
Discrete Mathematics				2	2		60	5
Set Theory				2	2		60	5

3RD YEAR OF STUDY, MODULE Communication Systems – CS

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Communication Networks	2	2					60	5
Network and Mobile Operating Systems	2	2					60	5
Social Network Analysis	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Introduction to Embedded Systems and the Internet of Things				2	2		60	5
Computer System Administration				2	2		60	5
Elective course				2	2		60	5
Elective course (common elective course offered by the University of Rijeka/course from another module/common elective course)							60	3
Undergraduate Internship							60	4
Bachelor’s Thesis							60	8
Total ECTS credits								**30+30**

ELECTIVE COURSES FOR THE MODULE Communication Systems – CS

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Programming Paradigms and Languages	2	2					60	5
Code Optimization	2	2					60	5
Web Programming	2	2					60	5
Multimedia Technologies	2	2					60	5
Computer Graphics	2	2					60	5

Computer Animation	2	2					60	5
Database Administration and Security	2	2					60	5
User Interface and Interaction Design	2	2					60	5
Specific Purpose Information Systems	2	2					60	5
System Dynamics	2	2					60	5
Programming for Data Science	2	2					60	5
Introduction to Theoretical Computing	2	2					60	5
Combinatorics	2	2					60	5
Numerical Mathematics	2	2					60	5
Mathematical Logic	2	2					60	5
Desktop and Mobile Application Development				2	2		60	5
Declarative Programming Languages				2	2		60	5
Fundamentals of Game Development				2	2		60	5
Multimedia Design				2	2		60	5
Introduction to Data Analysis and Visualization				2	2		60	5
Next Generation Databases				2	2		60	5
Programming for Complex Problem Solving				2	2		60	5
Discrete Mathematics				2	2		60	5
Set Theory				2	2		60	5

3RD YEAR OF STUDY, MODULE Multimedia Systems – MS

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Multimedia Technologies	2	2					60	5
Computer Graphics	2	2					60	5
Computer Animation	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Fundamentals of Game Development				2	2		60	5
Multimedia Design				2	2		60	5
Elective course				2	2		60	5
Elective course (common elective course offered by the University of Rijeka/course from another module/common elective course)							60	3
Undergraduate Internship							60	4
Bachelor’s Thesis							60	8
Total ECTS credits								**30+30**

ELECTIVE COURSES FOR THE MODULE Multimedia Systems – MS

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Programming Paradigms and Languages	2	2					60	5
Code Optimization	2	2					60	5
Web Programming	2	2					60	5
Communication Networks	2	2					60	5
Network and Mobile Operating Systems	2	2					60	5
Social Network Analysis	2	2					60	5
Database Administration and	2	2					60	5

Security								
User Interface and Interaction Design	2	2					60	5
Specific Purpose Information Systems	2	2					60	5
System Dynamics	2	2					60	5
Programming for Data Science	2	2					60	5
Introduction to Theoretical Computing	2	2					60	5
Combinatorics	2	2					60	5
Numerical Mathematics	2	2					60	5
Mathematical Logic	2	2					60	5
Desktop and Mobile Application Development				2	2		60	5
Declarative Programming Languages				2	2		60	5
Introduction to Embedded Systems and the Internet of Things				2	2		60	5
Computer System Administration				2	2		60	5
Introduction to Data Analysis and Visualization				2	2		60	5
Next Generation Databases				2	2		60	5
Programming for Complex Problem Solving				2	2		60	5
Discrete Mathematics				2	2		60	5
Set Theory				2	2		60	5

3RD YEAR OF STUDY, MODULE Information Systems – IS

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Database Administration and Security	2	2					60	5
User Interface and Interaction Design	2	2					60	5
Specific Purpose Information Systems	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Elective course	2	2					60	5
Introduction to Data Analysis and Visualization				2	2		60	5
Next Generation Databases				2	2		60	5
Elective course				2	2		60	5
Elective course (common elective course offered by the University of Rijeka/course from another module/common elective course)							60	3
Undergraduate Internship							60	4
Bachelor’s Thesis							60	8
Total ECTS credits								**30+30**

ELECTIVE COURSES FOR THE MODULE Information Systems – IS

	5th semester class hours/week			6th semester class hours/week			Total class hours	ECTS credits
COURSE	**L**	**E**	**S**	**L**	**E**	**S**		
Programming Paradigms and Languages	2	2					60	5
Code Optimization	2	2					60	5
Web Programming	2	2					60	5
Communication Networks	2	2					60	5
Network and Mobile	2	2					60	5

Operating Systems								
Social Network Analysis	2	2					60	5
Multimedia Technologies	2	2					60	5
Computer Graphics	2	2					60	5
Computer Animation	2	2					60	5
System Dynamics	2	2					60	5
Programming for Data Science	2	2					60	5
Introduction to Theoretical Computing	2	2					60	5
Combinatorics	2	2					60	5
Numerical Mathematics	2	2					60	5
Mathematical Logic	2	2					60	5
Desktop and Mobile Application Development				2	2		60	5
Declarative Programming Languages				2	2		60	5
Introduction to Embedded Systems and the Internet of Things				2	2		60	5
Computer System Administration				2	2		60	5
Fundamentals of Game Development				2	2		60	5
Multimedia Design				2	2		60	5
Programming for Complex Problem Solving				2	2		60	5
Discrete Mathematics				2	2		60	5
Set Theory				2	2		60	5

4. Conditions for carrying out the study programme

4.1. LOCATION FOR CARRYING OUT THE STUDY PROGRAMME

Since October 2012, the Department of Informatics, University of Rijeka has been operating in a building situated within the University Campus on Trsat, so the proposed study programme will be carried out at the same location.

4.2. INFORMATION ON FACILITIES AND EQUIPMENT FOR CARRYING OUT THE STUDY PROGRAMME

The Department of Informatics, situated within the Campus building intended for university departments, has at its disposal 14.86% of net surface area of the entire object, which amounts to 1,411.73 m^2. This includes:

- two lecture halls that can accommodate 150 and 100 students respectively, with an LCD projector and a computer,
- two lecture halls that can accommodate 40 to 50 students, with an LCD projector and a computer,
- three computer classrooms equipped with the total of 83 personal computers and an LCD projector (from the academic year 2019/2020, we are planning to equip an additional computer classroom with another 36 computers),
- two laboratories equipped with state-of-the-art computer equipment,
- a room used by doctoral students for their work and research,
- computer classroom for independent work and training, with 15 computers,
- a room intended for meetings and presentations with an LCD projector and a computer,
- 32 offices for lecturers and associates, equipped with computers,
- library with a reading room within the Department building, equipped with IT equipment.

Hallways, toilets, repositories and technical facilities are shared with other employees of the University of Rijeka operating within the building.

4.3. NAMES OF LECTURERS AND NUMBER OF ASSOCIATES

No.	Name of the course	Course instructor
1.	Mathematics 1	Asst. Prof. Milena Sošić, PhD
2.	Programming 1	Full Prof. Maja Matetić, PhD
3.	Fundamentals of Informatics	Full Prof. Sanda Martinčić-Ipšić, PhD
4.	Multimedia Systems	Full Prof. Nataša Hoić-Božić, PhD
5.	Fundamentals of Economics for IT students	Asst. Prof. Borna Debelić, PhD
6.	English Language for IT Profession	Irena Grubica, MSc / Lucia Načinović Prskalo, PhD
7.	Physical Education 1	
8.	Mathematics 2	Asst. Prof. Ana Jurasić, PhD
9.	Programming 2	Full Prof. Maja Matetić, PhD
10.	Operating Systems	Assoc. Prof. Božidar Kovačić, PhD
11.	Data Modelling	Full Prof. Mile Pavlić, PhD
12.	Computer Architecture and Organization	Full Prof. Ivo Ipšić, PhD
13	Fundamentals of Probability and Statistics	Asst. Prof. Davor Dragičević, PhD
14.	Physical Education 2	
15.	Mathematics 3	Asst. Prof. Marija Maksimović, PhD
16.	Analysis of Business Processes	Full Prof. Mile Pavlić, PhD
17.	Object-Oriented Programming	Assoc. Prof. Marina Ivašić-Kos, PhD
18.	Computer Networks	Full Prof. Mario Radovan, PhD / Vedran Miletić, PhD
19.	Databases	Full Prof. Patrizia Poščić, PhD
20.	Introduction to Software Engineering	Assoc. Prof. Sanja Čandrlić, PhD
21.	Operations Research	Asst. Prof. Martina Holenko Dlab, PhD
22.	Introduction to Web Programming	Full Prof. Mario Radovan, PhD / Igor Jugo, PhD
23.	Algorithms and Data Structures	Asst. Prof. Marija Brkić Bakarić, PhD
24.	Information Technology Project Management	Full Prof. Sanda Martinčić Ipšić, PhD
25.	Information Systems Development	Full Prof. Mile Pavlić, PhD
26.	Security of Information and Communication Systems	Assoc. Prof. Božidar Kovačić, PhD
27.	Programming Paradigms and Languages	Assoc. Prof. Marina Ivašić-Kos, PhD
28.	Code Optimization	Asst. Prof. Miran Pobar, PhD / Vedran Miletić, PhD
29.	Web Programming	Assoc. Prof. Božidar Kovačić, PhD / Igor Jugo, PhD
30.	Communication Networks	Full Prof. Mario Radovan, PhD / Vedran Miletić, PhD
31.	Network and Mobile Operating Systems	Assoc. Prof. Božidar Kovačić, PhD
32.	Social Network Analysis	Assoc. Prof. Ana Meštrović, PhD
33.	Multimedia Technologies	Asst. Prof. Miran Pobar, PhD
34.	Computer Graphics	Asst. Prof. Martina Holenko Dlab, PhD
35.	Computer Animation	Full Prof. Nataša Hoić-Božić, PhD / Vanja Slavuj, PhD
36.	Database Administration and Security	Full Prof. Patrizia Poščić, PhD
37.	User Interface and Interaction Design	Assoc. Prof. Sanja Čandrlić, PhD
38.	Specific Purpose Information Systems	Assoc. Prof. Sanja Čandrlić, PhD / Martina Ašenbrener Katić, PhD
39.	System Dynamics	Asst. Prof. Marija Brkić Bakarić, PhD
40.	Programming for Data Science	Full Prof. Maja Matetić, PhD

41.	Introduction to Theoretical Computing	Full Prof. Sanda Martinčić-Ipšić, PhD
42.	Combinatorics	Full Prof. Sanja Rukavina, PhD
43.	Numerical Mathematics	Asst. Prof. Bojan Crnković, PhD
44.	Mathematical Logic	Asst. Prof. Tajana Ban Kirigin, PhD
45.	Bachelor’s Thesis	Assoc. Prof. Ana Meštrović, PhD
46.	Undergraduate Internship	Asst. Prof. Martina Holenko Dlab, PhD
47.	Desktop and Mobile Application Development	Asst. Prof. Marija Brkić Bakarić, PhD
48.	Declarative Programming Languages	Assoc. Prof. Ana Meštrović, PhD
49.	Introduction to Embedded Systems and the Internet of Things	Asst. Prof. Miran Pobar, PhD
50.	Computer System Administration	Full Prof. Ivo Ipšić, PhD / Vedran Miletić, PhD
51.	Fundamentals of Game Development	Assoc. Prof. Marina Ivašić-Kos, PhD
52.	Multimedia Design	Full Prof. Nataša Hoić-Božić, PhD
53.	Introduction to Data Analysis and Visualization	Full Prof. Sanda Martinčić-Ipšić, PhD / Lucia Načinović Prskalo, PhD
54.	Next Generation Databases	Full Prof. Patrizia Poščić, PhD / Danijela Jakšić, PhD
55.	Programming for Complex Problem Solving	Assoc. Prof. Ana Meštrović, PhD
56.	Discrete Mathematics	Full Prof. Dean Crnković, PhD
57.	Set Theory	Asst. Prof. Tajana Ban Kirigin, PhD

Employees of the University Department of Informatics:

Full Prof. Mario Radovan, PhD
Full Prof. Ivo Ipšić, PhD
Full Prof. Mile Pavlić, PhD
Full Prof. Nataša Hoić-Božić, PhD
Full Prof. Sanda Martinčić-Ipšić, PhD
Full Prof. Maja Matetić, PhD
Full Prof. Patrizia Poščić, PhD
Assoc. Prof. Sanja Čandrlić, PhD
Assoc. Prof. Marina Ivašić-Kos, PhD
Assoc. Prof. Božidar Kovačić, PhD
Assoc. Prof. Ana Meštrović, PhD
Asst. Prof. Marija Brkić Bakarić, PhD
Asst. Prof. Martina Holenko Dlab, PhD
Asst. Prof. Miran Pobar, PhD
Martina Ašenbrener Katić, PhD
Danijela Jakšić, PhD
Igor Jugo, PhD
Vedran Miletić, PhD
Lucia Načinović Prskalo, PhD
Vanja Slavuj, PhD
Slobodan Beliga, MA in informatics and education in informatics
Ivona Franković, MA in informatics and education in informatics
Kristian Stančin, MA in informatics
Maja Vrancich, prof.
Marina Žunić, MA in education in informatics

Employees of the University Department of Mathematics:

Full Prof. Dean Crnković, PhD
Full Prof. Sanja Rukavina, PhD
Asst. Prof. Tajana Ban Kirigin, PhD
Asst. Prof. Bojan Crnković, PhD
Asst. Prof. Davor Dragičević, PhD
Asst. Prof. Ana Jurasić, PhD
Asst. Prof. Marija Maksimović, PhD
Asst. Prof. Milena Sošić, PhD

External associates:

Asst. Prof. Borna Debelić, PhD
Irena Grubica, MSc, senior lecturer

4.4. METHODS FOR MONITORING QUALITY AND PERFORMANCE OF THE STUDY PROGRAMME

Quality and performance of the Undergraduate study programme in informatics will be monitored in accordance with statutory regulations in the Republic of Croatia aimed at ensuring quality at higher education institutions, as well as with ordinances and standards prescribed at the level of the University of Rijeka and Department of Informatics.

Periodical evaluations will be made for the purpose of ensuring and continuously improving the quality of classes and study programme as part of the activities of the Quality Assurance Committee at the Department of Informatics.

Continuous evaluation will be carried out in the course of the study programme, based on the results obtained through:

- student class evaluation (anonymous survey) carried out at the end of each semester for each course and for the study programme as a whole,
- analysis of pass rate, i.e. number of students who were successful in individual courses (percentage of students who passed the course and their average grade),
- analysis of student enrolment in higher years of study,
- contacting students after finishing the study programme (alumni) and conducting surveys about students during internship and employed students among employers.

The evaluation methods above will enable the detection of potential problems that could cause low-quality, inefficient or prolonged study by individual students and, in consultation with students, the identification of possible causes, so that necessary steps could be taken (for example, introducing student tutoring for courses with low pass rates) to eliminate them.

In addition, with the aim of raising the quality of the study programme, continuous efforts will be made to facilitate the professional development of university professors who participate in the implementation of the study programme through various forms of education for improving their teaching competencies. It is also planned to conduct peer review among teaching staff.

With the aim of improving the quantity and quality of mandatory literature that should be available to students during their study, teaching staff will be encouraged to publish their works, and necessary financial support will be provided.

Teaching staff will be encouraged to use e-learning technologies and methods, so that e-learning tools could be used for part of the courses within the study programme, either in blended or completely online form.