University of Rijeka
Faculty of Engineering

CURRICULUM FOR
UNDERGRADUATE PROGRAM OF COMPUTING

Rijeka, December 2025

1. PLAN OF STUDY

1.1. The list of compulsory and elective courses with the number of active classes required for their performance and ECTS credits

1st semester								
	Subject title		**Hours / week**					**ECTS**
		L	**aT**	**lT**	**dT**	**S**	**∑**	
	Mathematics 1	3	3				6	7
	Digital Logic	2	2				4	6
	Electrical Engineering CE	3	2	1			6	8
	Programming 1	2		2			4	6
	Communication Skills	1	1				2	3
	TOTAL						**22**	**30**

L - lectures, aT – auditory tutorials, lT – laboratory tutorials, dT – design tutorials, S - seminars

2nd semester								
	Subject title		**Hours / week**					**ECTS**
		L	**aT**	**lT**	**dT**	**S**	**∑**	
	Mathematics 2	3	3				6	7
	Operating Systems 1	2		2			4	6
	Programming 2	3		2			5	7
	Computational Skills	2		1		1	4	6
	English Language in Engineering	1	2				3	4
	TOTAL						**22**	**30**

3rd semester								
	Subject title		**Hours / week**					**ECTS**
		L	**aT**	**lT**	**dT**	**S**	**∑**	
	Mathematics 3 CE	2	2				4	6
	Algorithms and Data Structures	2		2			4	6
	Computer Architecture	2	2				4	6
	Operating Systems 2	2		2			4	6
	Introduction to Object Oriented Programming	2		2			4	6
	TOTAL						**20**	**30**

4th semester								
	Subject title		Hours / week					ECTS
		L	aT	IT	dT	S	∑	
	Mathematics 4 CE	2	1	1			4	6
	Embedded Systems	2		2			4	6
	Numerical Methods	2		2			4	6
	Elective Subject 1						4	6
	Professional Practice 1							6
	TOTAL						15	30

Elective Subject 1								
	Subject title		Hours / week					ECTS
		L	aT	IT	dT	S	∑	
	Modelling of Process Information Systems	2		2			4	6
	Computer Simulations in Engineering	1		2		1	4	6
	Organization of Business Systems	2	2				4	6
	Robotic Systems	2		2			4	6

5th semester								
	Subject title		Hours / week					ECTS
		L	aT	IT	dT	S	∑	
	Computer Networks	2	1	1			4	6
	Database Systems	2		2			4	6
	Web Application Development	2		2			4	6
	Elective Subject 2						4	6
	Elective Project[1]				4		4	6
	TOTAL						20	30

[1] election from a list of offered groups of courses:

Elective project – group 1: Digital Logic, Introduction to Artificial Intelligence

Elective project – group 2: Database Systems, Introduction to Object Oriented Programming

Elective project – group 3: Computer Architecture, Computer Skills, Web Application Development

Elective project – group 4: Operating Systems 1/2, Computer Networks, Embedded Systems

Elective project – group 5: Numerical Methods, Computer Graphics

In principal, the teacher of the previously mentioned courses is the teacher of the group of courses.

Elective Subject 2								
	Subject title		Hours / week					ECTS
		L	aT	IT	dT	S	∑	
	Computer Aided Measurements	2		2			4	6
	Signals and Systems	3	1				4	6
	Laboratory of Embedded Systems	1		1		2	4	6

6th semester								
	Subject title		**Hours / week**					**ECTS**
		L	**aT**	**IT**	**dT**	**S**	**∑**	
	Software Engineering	2		2		1	5	6
	Computer Graphics	2		2			4	6
	Introduction to Artificial Intelligence	2		2			4	6
	Skill[2]						(3)	(3)*
	Bachelor Thesis							12
	TOTAL						**13**	**30**

[2] Skill represents an optional course which is not included in the 30 ECTS per semester count

Free Elective Subjects								
	Subject title		**Hours / week**					**ECTS**
		L	**aT**	**IT**	**dT**	**S**	**∑**	
	Agile Development of Digital Platforms			3			3	3
	React and .NET for Full-stack Application Development			3			3	3
	Computer Practicum					3	3	3
	Mobile Application Security			2		1	3	3

UNDERGRADUATE PROGRAM OF COMPUTING	**Hours**	**ECTS**
TOTAL	**113 (+3)**	**180**

1.2. Courses description

Basic description		
Course title	Agile Development of Digital Platforms	
Study programme	Undergraduate University Study of Computing	
Course status	optional	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	3
	Number of hours (L+E+S)	0+45+0

1. COURSE DESCRIPTION

1.1. Course objectives

Master the fundamental concepts of digital platforms, agile software development, and application architecture. Develop teamwork skills and work with code and database versioning and code review to improve team collaboration and code quality. Understand and apply the basics of building a monolithic application using the techniques of the SOLID principles, principles of clean code, authentication, validation and logging. Learn about *Azure Cloud* and how to deploy to *Azure Cloud* easily.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Understand the basic concepts and principles of digital platforms and the role of digital platforms in modern business.
Build monolithic applications using best practices in design and implementation.
Configure and deploy applications on the *Azure Cloud* platform.

1.4. Course content

Basic concepts of digital platforms, agile software development, and application architecture. Understanding and implementation of a monolithic application using SOLID principles, clean code principles, authentication, validation, and logging. Development of the data and API layers with an understanding of the business logic and versioning of the program code. Basic concepts of the *Azure Cloud* platform and application deployment.

1.5. Teaching methods	[ ] lectures	[X] individual assignment
	[ ] seminars and workshops	[ ] multimedia and network
	[X] exercises	[ ] laboratories
	[ ] long distance education	[ ] mentorship
	[ ] fieldwork	[ ] other ________________
1.6. Comments		

1.7. Student's obligations

Class attendance, making a project.

1.8. Evaluation of student's work

Course attendance	1.5	Activity/Participation		Seminar paper		Experimental work	

Written exam		Oral exam		Essay		Research	
Project	1.5	Continuous assessment		Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Delivery and presentation of the project.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008.

1.11. Optional / additional reading (at the time of proposing study programme)

Robert C. Martin: Agile Software Development, Principles, Patterns, and Practices, Prentice Hall, 2002.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008. Available online	*online copies*	-

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Algorithms and Data Structures**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquiring basic knowledge and understanding of simple and abstract data types, algorithm complexity analysis, sort and search algorithms. Developing the capability of solving complex programming problems.

1.2. Course enrolment requirements

Passed Programming II.

1.3. Expected course learning outcomes

Analyze the possibilities of using different data structures.
Formulate the implementation of basic algorithms for sorting and searching.
Recommend appropriate data structures and algorithms when solving specific problems.
Build software solutions for concrete problems based on the application of appropriate algorithms and data structures.
Propose software solutions based on existing libraries of algorithms and data structures.
Evaluate the time complexity of algorithms.

1.4. Course content

Introduction: problem solving, algorithm, pseudo code, data types, time complexity of algorithms. Abstract data type. List. Stack. Queue. Recursion and iteration. Sorting and searching algorithms. Trees. Graphs. Dynamic Programming. Hash tables.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other _______________
1.6. Comments		

1.7. Student's obligations

Midterm exams, Continuous assessment, written exam.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	

Project		Continuous assessment	3	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

1.10. Assigned reading (at the time of the submission of study programme proposal)

1.11. Optional / additional reading (at the time of proposing study programme)

Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein: Introduction to Algorithms Third Edition
Wikibooks: Data structures http://en.wikibooks.org/wiki/Data_Structures
Wikibooks: Algorithms http://en.wikibooks.org/wiki/Algorithms
Mark Allen Weiss: Data structures and algorithm analysis in C++ / Edition:3rd ed. Publication:Boston : Pearson, Addison Wesley, 2006
Robert Sedgewick: Algorithms in C, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms, Addison-Wesley Professional, 2001.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Bachelor Thesis**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	12
	Number of hours (L+E+S)	-

1. COURSE DESCRIPTION

1.1. Course objectives

The Bachelor Thesis is an individual assignment and verification of student expertises, which should show the appropriate level of engineering skills for individually solving specific professional task.

1.2. Course enrolment requirements

Enrolled course from which the Bachelor Thesis is selected.

1.3. Expected course learning outcomes

Apply acquired knowledge, expertises and skills of the content of Bachelor Thesis associated course. Solve practical task. Acquire competence for individually solving specific professional task.

1.4. Course content

The content of the Final Work is based on the application of acquired knowledge from educational programs at the undergraduate university studies. Final thesis can be specified from a particular course specific professional content and exceptionally from course that belongs to the group of shared content, when it represents a broader entity with a particular course specific content of the studies. Student enrollers the Final Work by enrolling the last semester. Thesis of the Final Work is establishes by Commission for Final Works, based on suggestion of teacher who will mentor the Final Work.

1.5. Teaching methods	[ ] lectures [ ] seminars and workshops [ ] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [X] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Attending the consultation, individually solving task and writing the Bachelor Thesis report.

1.8. Evaluation of student's work

Course attendance		Activity/Participation		Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment	9	Report	3	Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Assesses and evaluates the accuracy and completeness of a given task solving process, the Bachelor Thesis written report, and its oral presentation

1.10. Assigned reading (at the time of the submission of study programme proposal)

-

1.11. Optional / additional reading (at the time of proposing study programme)

-

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description

Course title	**Communication Skills**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	3
	Number of hours (L+E+S)	15+15+0

1. COURSE DESCRIPTION

1.1. Course objectives

The aim of the course is to enable students to acquire knowledge and skills related to the fundamental communication needs of engineers both in the domestic and international environment, such as presenting professional content, writing CVs, job applications, emails and reports in English and Croatian.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Apply the norms of the standard Croatian and English language in written and spoken public communication.
Apply the skills of asking questions and present professional content.
Apply the skills of writing official letters.
Critically assess one`s own and others` communication skills.
Negotiate and demonstrate the skill of assertive communication.
Actively participate in teamwork.

1.4. Course content

Introduction, active learning methods, learning styles. Verbal and non-verbal communication. Active listening. Questioning skills. Persuading and negotiation. Written communication: writing emails, CVs, job applications and reports. Presentation skills. Strategies for eliminating stage fright and fear of public speaking. Presentation of professional content. Communication and participation in group and teamwork. Critical assessment and providing feedback. Intercultural competence, cultural differences and etiquette.

1.5. Teaching methods

[X] lectures
[ ] seminars and workshops
[X] exercises
[ ] long distance education
[ ] fieldwork
[X] individual assignment
[ ] multimedia and network
[ ] laboratories
[ ] mentorship
[ ] other ________________

1.6. Comments

The course consists of: 1) lectures focused on a particular aspect of written and spoken public communication and 2) exercises in which the students solve specific language tasks from their field of profession, where they try to autonomously find and offer reasoned solutions, which are then commented on together, compared and valued.

1.7. Student's obligations

Course attendance, active participation in the teaching process, autonomous learning.

1.8. Evaluation of student's work

Course attendance	1	Activity/Participation		Seminar paper	0.5	Experimental work	
Written exam	0.5	Oral exam		Essay	0.5	Research	
Project		Continuous assessment	0.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, knowledge check (1 midterm test), preparing and giving a presentation, writing emails, a CV, a job application and a report.

1.10. Assigned reading (at the time of the submission of study programme proposal)

John W. Davies (2001), Communication skills. Pearson education Limited.
Mirjana Matea Kovač, Nina Sirković (2014), Presentation, writing and interpersonal communication skills, FESB

1.11. Optional / additional reading (at the time of proposing study programme)

Thomas E. Harris, John C. Sherblom (2018), Small Group and Team Communication, Waveland Press
Kamilo Antolović, Nikša Sviličić (2020.), Komunikacijske vještine. Verbalne i neverbalne utjecajne tehnike, K&K promocija, Zagreb

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
John W. Davies (2001), Communication skills. Pearson Education Limited.	*1*	*70*
Mirjana Matea Kovač, Nina Sirković (2014), Presentation, writing and interpersonal communication skills, FESB	*1*	*70*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	Computer Aided Measurements	
Study programme	Undergraduate University Study of Computing	
Course status	elective	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Computer Aided Measurements enable students to understand advantages and possibilities of electronic measurement instruments, to independently analyze measurement problem and to realize virtual instrumentation.

1.2. Course enrolment requirements

Electrical Engineering CE.

1.3. Expected course learning outcomes

Interpret and explain the concept of measurement uncertainty and apply the model of measurement uncertainty in simpler examples.
Describe the operation of measuring amplifiers.
Describe the impact of noise and interference on the measurement result and methods of reducing them.
Describe the transfer characteristics of A/D and D/A converters and the operation of different configurations of A/D converters and choose the optimal converter for a specific measurement problem.
Implement a virtual instrument.
Analyze individual characteristics of automated instrumentation.

1.4. Course content

Introduction to the measurement science. The international system of units. Measurement errors. Measurement uncertainty. Noise and interference. Measurement amplifiers. Analog-digital converters. Digital-analog converters. Oscilloscopes. Automated measurements. Microprocessors and microcontrollers in computerized instrumentation. Examples of computer aided measurements: 3D multisensor coordinate measuring machines and systems for 3D scanning-digitalization-measurements. Communication with measurement equipment. Basic configurations of computerized measurement systems. Virtual instrumentation. Software for development of measurement applications.

1.5. Teaching methods	[X] lectures	[X] individual assignment
	[ ] seminars and workshops	[ ] multimedia and network
	[X] exercises	[X] laboratories
	[ ] long distance education	[ ] mentorship
	[ ] fieldwork	[ ] other _______________

1.6. Comments	

1.7. Student's obligations

Course and laboratory practice attendance, seminar paper, activity during course lectures, studying.

1.8. Evaluation of student’s work

Course attendance	2	Activity/Participation		Seminar paper	1	Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practice	1
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Assessment and evaluation of student’s work will be based on Continuous assessments, laboratory practice and based on seminar paper or final exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Vujević, D., Ferković, B.: Basics of Electrical Engineering Measurements, I. i II. part, Školska knjiga, Zagreb, 1996. (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

Šantić, A.: Electronic Instrumentation, 3rd Edition, Školska knjiga, Zagreb, 1993. (in Croatian)
Coombs, C.F.Jr.: Electronic Instrument Handbook, McGraw-Hill, 2nd Edition, 1999.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Vujević, D., Ferković, B.: Basics of Electrical Engineering Measurements, I. i II. part, Školska knjiga, Zagreb, 1996. (in Croatian)	*8*	*40*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Computer Architecture**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Obtaining basic knowledge of computer hardware.

1.2. Course enrolment requirements

Programming I

1.3. Expected course learning outcomes

Determine the categories of computer architecture and the working principles of known versions of computer architecture.
Analyze the operation of parts (elements) of the computer system.
Valorize the possibilities of connecting the processor with memory resources and input-output circuitry.
Analyze the principle of microcomputer instruction execution.
Propose a procedure for solving communication problems between processors and input-output units.
Build a programming solution for the target architecture in assembly language.

1.4. Course content

Computer Architecture definition and classification. Historical overview of computer development. Turing machine. Coding data and operations in a computer. Model of von Neumann Computer Architecture. Control unit. Simple microprocessor model instruction execution. RICS and CISC architecture. Pipeline architecture of microprocessor. Computer Buses. Computer memory system and Cache memory. Memory organization and virtual memory system. Input/output control system. Interrupt handling techniques. Overview of 8, 16, 32 and 64 bits computer architecture.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments	-	

1.7. Student's obligations

Course attendance, activity, homework, studying.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	

Written exam	1.5	Oral exam		Essay		Research	
Project		Continuous assessment	2.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity, homework, continuous knowledge testing (two mid-term exams), written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Ribarić, S.: Computer Architecture, Architecture and Organisation of Computer Systems, Algebra d.o.o. 2011 (in Croatian).
Ribarić, S.: Computer Architecture RISC i CISC, Školska knjiga, Zagreb, 1996 (in Croatian).
Ribarić, S.: Advanced Microprocessor Architectures, Školska knjiga, Zagreb, 1997 (in Croatian).

1.11. Optional / additional reading (at the time of proposing study programme)

Ribarić, S.: Arhitektura mikroprocesora, Tehnička knjiga, Zagreb, 1988.
Peruško, U., Glavinić, V.: Digitalni sustavi, Školska knjiga Zagreb, 2005.
Hennessey, J.L., Patterson D.A.: Computer Organization and Design : The Hardware/Software Interface, Morgan Kauf. Pub., San Mateo, 2013.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Ribarić, S.: Computer Architecture, Architecture and Organisation of Computer Systems, Algebra d.o.o. 2011 (in Croatian).	2	50
Ribarić, S.: Computer Architecture RISC i CISC, Školska knjiga, Zagreb, 1996 (in Croatian).	1	50
Ribarić, S.: Advanced Microprocessor Architectures, Školska knjiga, Zagreb, 1997 (in Croatian).	5	50

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Computer Graphics**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

An overview of the basics in computer graphics. Understanding of existing and capability to develop smaller computer graphics software.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Proper interpretation of principles of projective geometry.
Classification and name basic traits of curves and surfaces in use in computer graphics.
Development of computer programs using OpenGL and/or similar libraries.
Development of computer programs which display 2d and 3d objects, curves and surfaces, light, color and material assignment to objects.

1.4. Course content

Review the basics of computer graphics. Ortographic and perspective transformations. Graphic primitives and transformations. Parametric display of curves and surfaces. Elemental differential geometry. Bicubic presentation of the surface. Modeling body geometry. Network display. Visualization with basic bodies. Models and procedures of shading, shading. Set objects in a 3d scene. Light, materials, animation.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments	-	

1.7. Student's obligations

Course attendance, individual assignments and exercises.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio		Exercises	1				

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, homework, Continuous assessment (two partial exams), oral and written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

M. Čupić, Ž. Mihajlović, Interactive Computer Graphics through Examples in OpenGL, Zagreb, 2011 (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

Penna M. A., Patterson R. R., Projective geometry and its applications to computer graphics, Prentice-Hall, Englewood Cliffs, New Yersey
Yamagochy F., Curves and surfaces in computer aided geometric design, Springer-Verlag 1988.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
M. Čupić, Ž. Mihajlović, Interactive Computer Graphics through Examples in OpenGL, Zagreb, 2011 (in Croatian)	30	27

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Computer Networks**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Description and classification of the structure and architecture of computer networks and communication services. Defining the principles of operation of computer networks. Developing the ability to use basic network protocols and Internet services.

1.2. Course enrolment requirements

-

1.3. Expected course learning outcomes

Defining the concept, architecture, and organization of communication networks.
Explaining the operation mode and functions of communication networks.
Assessing protocol functions and services, as well as protocol interactions to select appropriate solutions.
Evaluating security threats and issues in building secure computer networks.
Applying Internet application layer protocols using appropriate implemented solutions.
Implementing simple network protocols.
Selecting basic configurations for network devices.

1.4. Course content

Organization of computer networks. OSI reference model. Physical layer for data transmission: theoretical foundations, transmission media. Building the physical layer, cabling. Data link layer. Error detection and correction, examples of protocols, data link layer on the Internet. Media Access Control (MAC) sublayer, channel allocation for transmission. IEEE 802 LAN standards for LANs. Network layer. Routing algorithms and congestion control. Connecting networks. Network layer on the Internet. Transport layer services and elements of transport protocols. Internet transport layer. Application layer. Internet applications and their protocols. Application of computer networks. Network security.

1.5. Teaching methods	[X] lectures	[X] individual assignment
	[ ] seminars and workshops	[ ] multimedia and network
	[X] exercises	[X] laboratories
	[ ] long distance education	[ ] mentorship
	[ ] fieldwork	[ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity/participation, studying, teamwork.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1	Oral exam	1	Essay		Research	
Project		Continuous assessment	1	Report		Practice	1
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Midterm exams, laboratory skills assessments, homework, final oral and/or written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Kurose, J.F., Ross K.W.: Computer Networking: A Top-Down Approach, 86th Edition, Pearson Education, 2021

1.11. Optional / additional reading (at the time of proposing study programme)

Peterson, L., Davie, B.:Computer Networks, Fifth Edition: A Systems Approach, Morgan Kaufmann, 2020
Radovan, M.: Računalne mreže (1), Rijeka, Digital point tiskara, 2010.
Radovan, M.: Računalne mreže (2), Rijeka, Digital point tiskara, 2011.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Kurose, J.F., Ross K.W.: Computer Networking: A Top-Down Approach, 8th Edition, Pearson Education, 2021	1	90

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Computer Practicum**	
Study programme	**Undergraduate University Study of Computing**	
Course status	optional	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	3
	Number of hours (L+E+S)	0+45+0

1. COURSE DESCRIPTION

1.1. Course objectives

Independent modelling of the system and implementing the program code with an argumentation of the obtained results.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Analyze the given process for the purposes of creating a computer model.
Create computer code and design the user interface.
Analyze the computer system.
Present the obtained results in an argumentative manner.

1.4. Course content

Analytical understanding of the given process for the needs of creating a computer model. Creation of computer code and user interface. Analysis of the obtained solutions of the simulation system. Presentation of the obtained results.

1.5. Teaching methods

[ ] lectures
[X] seminars and workshops
[ ] exercises
[ ] long distance education
[ ] fieldwork

[X] individual assignment
[ ] multimedia and network
[ ] laboratories
[X] mentorship
[ ] other ________________

1.6. Comments

1.7. Student's obligations

Creation, presentation and defense of the project assignment.

1.8. Evaluation of student's work

Course attendance		Activity/Participation		Seminar paper	1	Experimental work	1
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment		Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Assessment and evaluation of students' work will be based on the creation, presentation and defense of the project assignment.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Letaw, Lara. Handbook of Software Engineering Methods (2nd Edition). Oregon State University, 2024. Available online: https://open.oregonstate.education/setextbook/

1.11. Optional / additional reading (at the time of proposing study programme)

-

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Letaw, Lara. Handbook of Software Engineering Methods (2nd Edition). Oregon State University, 2024. Available online: https://open.oregonstate.education/setextbook/	online copies	

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Computer Simulations in Engineering**	
Study programme	**Undergraduate University Study of Computing**	
Course status	elective	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	15+30+15

1. COURSE DESCRIPTION

1.1. Course objectives

General knowledge of computer simulation technology. Understanding the basis of mathematical modeling. Knowing capabilities and limitations of computer simulations. Identifying methods for solving engineering problems using computer simulations.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Correctly explain the methodology of mathematical modeling and the application of numerical methods in computer simulations.
Classify main types of numerical meshes.
Explain the integral process of applying computer simulations in solving engineering problems.
Give examples of methods and tools for numerical modeling.
Perform simulations in a given software of simple problems of continuum mechanics.

1.4. Course content

Review of existing CAE systems. The process of mathematical modeling. Using finite-element model of solid body mechanics. Using computational fluid dynamics. Modeling of heat transfer. Introduction to commercial software and I-DEAS, CATIA, FLUENT. Structured and unstructured mesh, boundary condition definitions. Understanding the entire process of application of computer simulation for solving engineering problems.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Attendance, class participation, individual assignment.

1.8. Evaluation of student's work

Course attendance	1,5	Activity/Participation		Seminar paper	3	Experimental work	
Written exam		Oral exam		Essay		Research	

Project		Continuous assessment	1,5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity, continuous knowledge testing, seminar paper.

1.10. Assigned reading (at the time of the submission of study programme proposal)

I-DEAS, CATIA, FLUENT User Manuals.

1.11. Optional / additional reading (at the time of proposing study programme)

-

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
I-DEAS, CATIA, FLUENT User Manuals.	online copies	50

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Computer Skills**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+15+15

1. COURSE DESCRIPTION

1.1. Course objectives

Getting acquainted with software environments for engineering and scientific computation, professional text- and bibliography- processing. Getting familiar with typical elements of project proposal preparation and software solutions to support it.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

To use tools for engineering and scientific computation and professional processing of text and bibliography. To get familiar with some software tools for teamwork and communication. To learn key elements of a project proposal. To create a simple project proposal comprising a work plan, budget, and time plan.

1.4. Course content

Cloud-based spreadsheet calculations. Practical usage of a high-level programming language (basic syntax, arrays, matrices, solving a system of linear equations, polynomials, functional programming, root finding, plots, animations, derivatives, integrations). Usage of reactive notebooks. Software-supported classification and storage of bibliography units. A cloud-based preparation of a complex text (formatting of headings, text, figures, tables, citations, equations, and bibliography). Overview and basic practical usage of software tools for team collaboration and communication. Typical elements of a project proposal with an emphasis on the work plan, financial plan (budget), and the time plan (the Gantt chart).

1.5. Teaching methods	[X] lectures [X] seminars and workshops [X] exercises [X] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Minimum of 80% class attendance and fair participation in team assignments.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper	1	Experimental work	
Written exam	1	Oral exam		Essay		Research	

Project	2	Continuous assessment		Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Projects, seminars, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Course notes.

1.11. Optional / additional reading (at the time of proposing study programme)

Handbooks that support learning of the respective programming language.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
N. Kalicharan, Julia – Bit by Bit: Programming for Beginners, Springer, 1st ed. 2021.	10	70
D. J. Zea, Interactive Visualization and Plotting with Julia: Create impressive data visualizations through Julia packages such as Plots, Makie, Gadfly, and more, Packt Publishing, 2022.	10	70

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Database Systems**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student ‘s workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Understanding database management systems. Database design. Defining relational databases and handling data. Enforcing data integrity and data protection. Using software tools for designing and building databases, and for data management.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Determine the basic concepts of data, information and databases.
Determine the components of a database management system.
Select and apply the database modelling principles (ER model, functional dependencies, normalization).
Understand relational algebra and the syntax and semantics of the SQL.
Recommend queries in SQL when implementing a database, when working with databases, and when solving new tasks.
Understand functions, stored procedures, and triggers when working with databases.
Formulate the basic principles of database protection.

1.4. Course content

Basic concepts of database and database management systems. Data models. Relational algebra and relational model. Logical design of databases. Entity-relationship model. Transforming entity-relationship model into relations. Functional dependencies and normalization. Structured Query Language (SQL). Data integrity and security. Transactions.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[ ] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student’s obligations

Class attendance, attending tests, solving tasks independently

1.8. Evaluation of student’s work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	

Written exam	1.5	Oral exam		Essay		Research	
Project		Continuous assessment	2.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Tests, laboratory exercises, written exam

1.10. Assigned reading (at the time of the submission of study programme proposal)

Thomas M. Connolly, Carolyn E. Begg: Database Systems – A Practical Approach to Design, Implementation and Management (6th Edition), Pearson Education, 2015.

Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: Database Systems – The Complete Book (2nd Edition), Pearson Education, 2009.

1.11. Optional / additional reading (at the time of proposing study programme)

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Thomas M. Connolly, Carolyn E. Begg: Database Systems – A Practical Approach to Design, Implementation and Management (6th Edition), Pearson Education, 2015.	*1*	
Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: Database Systems – The Complete Book (2nd Edition), Pearson Education, 2009.	*1*	

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Digital Logic**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Understanding basic concepts of digital logic and operation of logic circuits. Understanding basic methods for analysing and designing combinational and sequential digital circuits and systems. Developing the ability of analysing, synthesizing and solving problems in the field of digital logic.

1.2. Course enrolment requirements

-

1.3. Expected course learning outcomes

Defining logical levels and basic characteristics of digitals signals. Applying various number systems. Using various codes to express digital data. Defining the Boolean algebra axioms and basic theorems. Knowing minimizition poredures for logical functions. Distinguishing AND-OR, AND-OR complement, XOR and NOR logic. Using various combinational logic circuits and functions. Explaining operational principles and applications of sequential logic circuits

1.4. Course content

To choose the appropriate level of standard combinational and sequential components in order to design simple digital circuits.
To design combinational and sequential digital circuits.
To apply Boolean algebra as a formal apparatus for describing combinational and sequential digital circuits.
To apply logic function minimization procedures.
To design logic functions for application in a programmable logic assembly.
To evaluate the results of modeling and synthesis of digital systems.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments	-	

1.7. Student's obligations

Lecture attendance, exercises attendance, individual studying.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation	1,5	Seminar paper		Experimental work	

Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	1,5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Continuous assessment, project, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

A. P. Godse and D. A. Godse: Digital Logic Circuits, Technical Publications, 2011.
U. Peruško i V. Glavinić: Digital Systems, Školska knjiga, 2005. (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

T. L. Floyd: Digital Fundamentals, 10/E, Prentice Hall, 2009.
M. M. Mano and M. D. Ciletti: Digital Design, 4/E, Prentice Hall, 2007.
W. Kleitz: Digital Electronics with VHDL, Prentice Hall, 2006.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
A. P. Godse and D. A. Godse: Digital Logic Circuits, Technical Publications, 2011.	*1*	*60*
U. Peruško i V. Glavinić: Digital Systems, Školska knjiga, 2005. (in Croatian)	*5*	*60*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Elective Project**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student ‘s workload coefficient	6
	Number of hours (L+E+S)	0+60+0

1. COURSE DESCRIPTION

1.1. Course objectives

Application of acquired knowledge and skills to solve practical problems in the field of associated course from which the project is elected.

1.2. Course enrolment requirements

Enrolled course from which the project is elected.

1.3. Expected course learning outcomes

Apply the knowledge and skills from professional content of the associated course. Solve practical task. Acquire competence for individually solving specific professional tasks.

1.4. Course content

Chosen chapter of associated course from which the project was elected.

1.5. Teaching methods	[ ] lectures [ ] seminars and workshops [ ] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [X] mentorship [ ] other ________________
1.6. Comments		

1.7. Student’s obligations

Attending the consultation, individually solving task and writing the project report.

1.8. Evaluation of student’s work

Course attendance		Activity/Participation		Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	3	Continuous assessment		Report		Practice	
Portfolio		Individual task solving	3				

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Assesses and evaluates the accuracy and completeness of the project task solution and its presentation.

1.10. Assigned reading (at the time of the submission of study programme proposal)

References listed for the associated course from which the project is elected.

1.11. Optional / additional reading (at the time of proposing study programme)

References listed for the associated course from which the project is elected.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	Electrical Engineering CE	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	8
	Number of hours (L+E+S)	45+45+0

1. COURSE DESCRIPTION

1.1. Course objectives

Mastering basic concepts, postulates and methods of electrostatics, magnetostatics and electrical circuits. Describing behavior of electromagnetic circuits' main components and analysis of electrical circuits.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Describe and apply basic electrostatics laws.
DC electrical circuit analysis.
Describe and apply basic electromagnetics laws.
AC electrical circuit analysis.
Operation of basic electronic devices validating.
Planning of measurement in electrical circuits.

1.4. Course content

Electrostatics - basic concepts and laws. Dielectric materials. Basic concepts and laws of DC circuits. DC circuit analysis - methods and theorems. Magnetostatics - basic concepts and laws. Magnetic materials and circuits. Basic concepts and laws of AC circuits. Operation principles of basic electronic devices.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity, studying.

1.8. Evaluation of student's work

Course attendance	3	Activity/Participation		Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	3	Report		Practice	1
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity, continuous knowledge testing (two mid-term exams), laboratory exercises, written and oral exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

N. Stojković, V. Sučić, S. Vlahinić, *Osnove elektrotehnike I*, Tehnički fakultet Sveučilišta u Rijeci i Fintrade, Rijeka, 2007.

N. Stojković, S. Vlahinić, V. Sučić, *Osnove elektrotehnike II*, Tehnički fakultet Sveučilišta u Rijeci i Fintrade, Rijeka, 2007.

1.11. Optional / additional reading (at the time of proposing study programme)

Pinter, V.: Fundamentals of electrical engineering – part I, Tehnička knjiga, Zagreb,1989, (in Croatian)
Pinter, V.: Fundamentals of electrical engineering – part II, Tehnička knjiga, Zagreb,1989, (in Croatian)

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
N. Stojković, V. Sučić, S. Vlahinić, *Osnove elektrotehnike I*, Tehnički fakultet Sveučilišta u Rijeci i Fintrade, Rijeka, 2007.	*82*	*82*
N. Stojković, S. Vlahinić, V. Sučić, *Osnove elektrotehnike II*, Tehnički fakultet Sveučilišta u Rijeci i Fintrade, Rijeka, 2007.	*82*	*82*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Embedded Systems**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Understanding the architecture and usage of microcontrollers. Understanding the principles and concepts of programming embedded systems. Acquiring knowledge and practical experience in implementing hardware and software components of embedded computer systems.

1.2. Course enrolment requirements

-

1.3. Expected course learning outcomes

Distinguishing components and applications in embedded systems.
Identifying basic components of embedded processors.
Analyzing embedded systems.
Using peripheral units of microcontrollers.
Applying procedures and tools for configuring and programming embedded systems.
Implementing and verifying the operation of algorithms for solving specific problems in embedded system applications.

1.4. Course content

Overview of applications using embedded systems. Architectures of processors in embedded systems. Components of microcontrollers: CPU cores, internal and external buses, interrupt system. Embedded peripheral units: I/O for general and specialized purposes, timer/counters, A/D and D/A units, units for serial communication. System integrity verification units. Programming embedded computer systems. Interrupt-driven operation concept. Specifics and challenges in hardware and firmware development in applications of embedded systems.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity/participation, studying, teamwork.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	0.5	Oral exam	0.5	Essay		Research	
Project		Continuous assessment	1	Report		Practice	2
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Lab quizzes, mid-term exams, homework, final exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Williams E.: AVR Programming: Learning to Write Software for Hardware, Make Community, 2014
Digan I.: Nucleo Boards Programming with the STM32CubeIDE: Hands-on in more than 50 projects, 2021

1.11. Optional / additional reading (at the time of proposing study programme)

Boxall J.: AVR Workshop: A Hands-On Introduction with 60 Projects, No Starch Press, 2022
Tahir M.: ARM Microprocessor Systems, CRC Press , 2020

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Williams E.: AVR Programming: Learning to Write Software for Hardware, Make Community, 2014	0	36
Digan I.: Nucleo Boards Programming with the STM32CubeIDE: Hands-on in more than 50 projects, 2021	*0*	*36*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**English Language in Engineering**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	4
	Number of hours (L+E+S)	15+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

The aim of this course is to enable students to acquire knowledge and skills related to the basic communication needs of engineers and to use the language of the engineering profession at an elementary level, respecting all four language skills: reading, listening, speaking and writing, all according to the Common European Framework of Reference for Languages (B1–B2 level).

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Students should be able to explain and compare general English with technical English on the basis of selected texts and topics.
Implement grammatical structures and aspects in written and oral exercises.
Recognise terminology, key words and/or information in selected texts as well as differentiate and analyse relevant elements in them.
Present the advantages and disadvantages in covered units. Express one's point of view and evaluate solutions of given problems both in oral and written form.

1.4. Course content

Topics: Engineering profession. Mathematical expressions and formulae. Computer essentials. Types of computers. Buying a computer. Operating Systems and Graphical User Interface. Application Programmes. Networks. The Internet. Internet Service Providers. Globalisation. Computers in Engineering. Data Security.
Specific grammatical and language structures (tenses, passive, articles and nouns, word formation, relative clauses, participles, etc.) will be covered within single units.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, active participation in the teaching process, autonomous learning.

1.8. Evaluation of student's work

Course attendance	1	Activity/Participation	1	Seminar paper	0.5	Experimental work	
Written exam	0.5	Oral exam		Essay		Research	
Project		Continuous assessment	1	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity in class, continuous evaluation of knowledge (two tests), seminar paper, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Velčić Janjetić, E.: Radni materijal za engleski jezik u inženjerstvu, 2024.

1.11. Optional / additional reading (at the time of proposing study programme)

Esteras, S. R. & Fabré, E. M.: Professional English in Use. ICT for Computers and the Internet. Cambridge University Press 2018.
Esteras, S. R.: Infotech. English for Computer Users. Cambridge University Press 2008.
Hill, D.: English for Information Technology 2. Pearson Education Limited 2017.
Glendinning E./McEwan J.: Oxford English for Information Technology (2. izdanje) Oxford University Press 2006.
Ibbotson, M.: Cambridge English for Engineering. Cambridge University Press 2015.
Paterson, K. & Wedge, R.: Oxford Grammar for EAP. Oxford University Press 2013.
Swan, M. & Walter, C.: Oxford English Grammar Course. Intermediate. Oxford University Press 2017.
Paterson, K. & Wedge, R.: Oxford Grammar for EAP. Oxford University Press 2013.
Vince M: Intermediate Language Practice, Heinemann Elt, Oxford 1998.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Velčić Janjetić, E.: Radni materijal za engleski jezik inženjerstvu, 2024.	*72*	*72*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Introduction to Artificial Intelligence**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Obtaining theoretical knowledge and developing skills to solve practical problems in the field of artificial intelligence. Acquiring the knowledge required for independent use of computing systems and software packages for solving common problems.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Pose a problem to be solved using artificial intelligence techniques.
Choose and apply an appropriate artificial intelligence technique to solve a specific problem.
Evaluate solutions obtained by applying artificial intelligence techniques to a specific problem.
Critically judge state space search methods, decision-making methods in (un)certainty and graphical models.

1.4. Course content

Introduction to artificial intelligence and application examples. State space search, informed search and adversarial search. Markov decision process. Reinforcement learning. Probability and inference. Bayesian network. Markov model and hidden Markov model. Machine learning basics.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other _______________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity in class, studying.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	2	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, midterm exams, exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Russell, S.J., Norvig P., Artificial Intelligence: A Modern Approach, 3rd ed., Pearson Education Limited, 2016

1.11. Optional / additional reading (at the time of proposing study programme)

Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.
Poole, David L., and Alan K. Mackworth. Artificial Intelligence: foundations of computational agents. Cambridge University Press, 2010.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Russell, S.J., Norvig P., Artificial Intelligence: A Modern Approach, 3rd ed., Pearson Education Limited, 2016	*3*	*60*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Introduction to Object Oriented Programming**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student ‘s workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Basic knowledge and skills for object oriented programming.

1.2. Course enrolment requirements

Passed Programming 1

1.3. Expected course learning outcomes

Explain the basic principles of the object-oriented paradigm.
Correctly interpret the concepts of class, object and interface.
Apply exception handling mechanism.
Correctly apply the concepts of abstraction, data encapsulation, inheritance and polymorphism for software development.
Use object-oriented programming language and associated libraries for program development.
Document the implemented software solution.
Test the behavior of the built program and remove the observed errors.
Apply the object-oriented design principles for the software solution.
Demonstrate the use of an integrated development framework for the development of an object-oriented software solution.

1.4. Course content

Object oriented programming using Java. Basic principles of object oriented programming, class and object, access control, inheritance and polymorphism, abstraction and interfaces, exceptions, input-output data streams, testing, memory, documentation.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student’s obligations

Course attendance, activity, studying, exercising.

1.8. Evaluation of student’s work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1.5	Oral exam		Essay		Research	
Project		Continuous assessment	2.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, continuous knowledge testing, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Herbert Schildt, Java: The Complete Reference, Tenth Edition, McGraw Hill Professional, 2017.
Java Tutorial
Java API

1.11. Optional / additional reading (at the time of proposing study programme)

Marko Čupić, Programiranje u Javi, FER, 2015
G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Languague User Guide, Addison – Wesley, 1998.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Java Tutorial, available at http://docs.oracle.com/javase/tutorial/index.html	Free access	
Java documentation, available at https://docs.oracle.com/javase/8/	Free access	

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Laboratory of Embedded Systems**	
Study programme	**Undergraduate University Study of Computing**	
Course status	elective	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	15+15+30

1. COURSE DESCRIPTION

1.1. Course objectives

Acquiring practical experience in designing and developing embedded systems through working on a specific project task.

1.2. Course enrolment requirements

Embedded systems

1.3. Expected course learning outcomes

Defining and utilizing components of embedded systems.
Analyzing functionality of embedded systems.
Designing complete embedded systems.
Utilizing features of embedded processors.
Debugging errors in the operation of embedded systems.

1.4. Course content

Hardware components of embedded systems.
Initialization of embedded processors.
Programming embedded systems and using development tools.
Debugging.
Solving project tasks.
Designing printed circuit boards and building embedded computers.

1.5. Teaching methods

[X] lectures
[X] seminars and workshops
[X] exercises
[ ] long distance education
[ ] fieldwork

[X] individual assignment
[ ] multimedia and network
[X] laboratories
[ ] mentorship
[ ] other _______________

1.6. Comments

1.7. Student's obligations

Course attendance, activity/participation, studying, teamwork.

1.8. Evaluation of student's work

Course attendance	1	Activity/Participation		Seminar paper		Experimental work	
Written exam	0.5	Oral exam	0.5	Essay		Research	1

Project	2	Continuous assessment		Report		Practice	1
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Exams, project task evaluation.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Digan I.: Nucleo Boards Programming with the STM32CubeIDE: Hands-on in more than 50 projects, 2021

1.11. Optional / additional reading (at the time of proposing study programme)

Martin T.: The Designer's Guide to the Cortex-M Processor Family, Newnes, 2022
Lutenberg A.: A Beginner’s Guide to Designing Embedded System Applications on Arm Cortex-M Microcontrollers, Arm Education Media, 2022
Tahir M.: ARM Microprocessor Systems, CRC Press , 2020

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Digan I.: Nucleo Boards Programming with the STM32CubeIDE: Hands-on in more than 50 projects, 2021	0	

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Mathematics 1**	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	7
	Number of hours (L+E+S)	45+45+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquiring basic knowledge and skills in linear algebra and calculus.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Solve matrix equations.
Solve a system of linear equations and analyze its solutions.
Present and apply the basics of vector algebra in V3.
Apply vector calculus in analytic spatial geometry.
Analyze the mutual position of two planes and the position of the direction and the plane in space.
Check the solutions of a linear system with three unknowns in the context of points, lines and planes in space.
Define and correctly interpret the basic concepts of functions of one variable.
Calculate limits and derivatives of functions of one variable.
Interpret derivatives of functions: mathematically, geometrically and physically.
Formulate the basic theorems of differential calculus.
Present properties of integrals, methods of integration and basic theorems of integral calculus.
Calculate indefinite and definite integrals.

1.4. Course content

Solving systems of linear equations. Matrices. Determinants.
Vectors and analytical geometry in space.
Single-variable functions. Limit values and continuous functions. Elementary functions.
Derivatives. Indefinite and definite integrals.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity/participation, studying.

1.8. Evaluation of student's work

Course attendance	3	Activity/Participation		Seminar paper		Experimental work	
Written exam	1.5	Oral exam		Essay		Research	
Project		Continuous assessment	2.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity/participation, Continuous assessment (mid-term exams, tests), and written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Slapničar I.: Mathematics 1, Sveučilište u Splitu FESB, Split 2002, online book (in Croatian)
Slapničar I.: Mathematics 1 – Workbook, Sveučilište u Splitu FESB, Split 2010, online book , (in Croatian)
Jurasić, K.-Dražić, I.: Mathematics I, Workbook, Tehnički fakultet, Rijeka, 2008. (in Croatian)
Štefan Trubić M., Sopta L., Črnjarić-Žic N., Maćešić S.: Mathematics, a collection of tasks: integrals, ordinary differential equations, functions of several variables, Rijeka 2012, (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

Elezović N., Aglić A., Linear algebra - a collection of tasks, Element, Zagreb 1999 (in Croatian)
Zill D., Wright W., Calculus: early transendentals, 4th edition, Jones and Bartlett publishers, 2011.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Slapničar I.: Mathematics 1, Sveučilište u Splitu FESB, Split 2002, online book (in Croatian)	72	72
Slapničar I.: Mathematics 1 – Workbook, Sveučilište u Splitu FESB, Split 2010, online book , (in Croatian)	72	72
Jurasić, K.-Dražić, I.: Mathematics I, Workbook, Tehnički fakultet, Rijeka, 2008. (in Croatian)	18	72
Štefan Trubić M., Sopta L., Črnjarić-Žic N., Maćešić S.: Mathematics, a collection of tasks: integrals, ordinary differential equations, functions of several variables, Rijeka 2012, (in Croatian)	20	72

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Mathematics 2**	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	7
	Number of hours (L+E+S)	45+45+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquiring basic knowledge and skills in application of calculus for single-variable functions, calculus for multi-variable functions, and ordinary differential equations.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Correctly interpret and apply the differential and integral calculus of functions of one variable.
Analyze the graph of a real function of one variable.
Apply the Taylor polynomial to approximate functions.
Apply integrals to calculate the area of plane sets, the length of the arc of a curve, the surface area and volume of rotating solids.
Introduce the basic theorems of differential calculus with several variables.
Calculate partial derivatives, differentials, derivatives and higher order differentials.
Apply Taylor's formula for functions of several variables to approximate function values.
Calculate and analyze extremes of functions of several variables and conditional extremes.
Apply double integrals to calculate surface areas and volumes and mass of solids.
Recognize the type of differential equation from the form of the equation.
Solve some first order differential equations and higher order linear differential equations.
Evaluate the theory of differential equations for problems from the profession.
Demonstrate mathematical modeling and problem solving skills using mathematical analysis methods in engineering practice.

1.4. Course content

Applications of single-variable calculus.
Multi-variable functions. Partial derivatives, differential calculus for two-variable functions and applications (approximations, local extremes, constrained optimization, optimization problems).
Double integral and applications.
First order ODE. Higher order ODE.
Systems of ODE. Applications of ODE.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________

1.6. Comments	

1.7. Student's obligations

Course attendance, activity/participation, studying.

1.8. Evaluation of student's work

Course attendance	3	Activity/Participation		Seminar paper		Experimental work	
Written exam	1.5	Oral exam		Essay		Research	
Project		Continuous assessment	2.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity/participation, Continuous assessment (mid-term exams, tests), and written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Slapničar I.: Mathematics 2, Sveučilište u Splitu FESB, Split 2002, online book, (in Croatian)
Štefan Trubić M., Sopta L., Črnjarić-Žic N., Maćešić S.: Mathematics, a collection of tasks: integrals, ordinary differential equations, functions of several variables, Rijeka 2012, (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

Kreyszig E., Advanced Engineering Mathematics, John Wiley & Sons, Inc., 1993.
Zill D., Wright W., Calculus: early transendentals, 4th edition, Jones and Bartlett publishers, 2011.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Slapničar I.: Mathematics 2, Sveučilište u Splitu FESB, Split 2002, online book, (in Croatian)	72	72
Štefan Trubić M., Sopta L., Črnjarić-Žic N., Maćešić S.: Mathematics, a collection of tasks: integrals, ordinary differential equations, functions of several variables, Rijeka 2012, (in Croatian)	20	72

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Mathematics 3 CE**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquisition of basic concepts from graph theory, understanding of selected algorithms from graph theory. Acquisition of knowledge and skills in combinatorics. Understanding of recursive relations and the structure of their solutions. Acquisition of basic concepts from probability theory and descriptive statistics and understanding of their application. Acquire knowledge and skills in Laplace transforms

1.2. Course enrolment requirements

Passed Mathematics 1.

1.3. Expected course learning outcomes

Formulate basic combinatorial structures and apply them to practical problems. Solve homogeneous and inhomogeneous linear recursions.
Formulate basic concepts from graph theory. Propose solutions to suitable engineering problems using selected algorithms from graph theory.
Formulate the basic concepts of descriptive statistics and analyze a set of statistical data. Formulate the concept of a random event, operations with events and the probability of a random event. Calculate the probabilities of certain events. Formulate Bayes' theorem and apply Bayes' formula.
Formulate and correctly interpret the concept of a discrete random variable. Be able to explain and calculate numerical indicators for discrete random variables. Describe the basic discrete distributions, explain their meaning and apply them in typical experiments.
Demonstrate the ability to mathematically model and solve problems using methods of discrete mathematics and probability in engineering practice. Calculate the Laplace transform of the given function.
Apply the Laplace transforms when solving differential equations, integral equations and systems of differential equations.

1.4. Course content

Discrete mathematics: Basic concepts of graph theory. Introduction to combinatorics. Recursions.
Basics of probability calculus: Descriptive statistics. The concept of a random event. The probability of a random event. Bayes formula. Discrete random variables.
Laplace transform: properties and applications.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[ ] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity, mid-term exams, tests.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1.5	Oral exam	0.5	Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity, continuous knowledge testing (mid-term exams, quizzes, tests), written exam, oral exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Elezović, N.: Fourier series and integral, Laplace transform, (FER) Biblioteka Bolonja, Element, 2006. (in Croatian)
Žubrinić D.: Introduction to discrete mathematics, Biblioteka Bolonja, Element, Zagreb 2006. (in Croatian)
Pavčević M., Introduction to graph theory, Biblioteka Bolonja, Element, Zagreb 2006. (in Croatian)
Elezović, N.: Discrete probability, Element, Zagreb, 2008. (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

Črnjarić-Žic N.: Internal lecture notes about engineering statistics.
Kreyszig, E.: Advanced Engineering Mathematics, John Wiley & Sons, Inc., 1993.
Črnjarić-Žic N., Štefan Trubić M., Internal lecture notes about Laplace transforms.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Elezović, N.: Fourier series and integral, Laplace transform, (FER) Biblioteka Bolonja, Element, 2006. (in Croatian)	10	70
Žubrinić D.: Introduction to discrete mathematics, Biblioteka Bolonja, Element, Zagreb 2006. (in Croatian)	4	70
Pavčević M., Introduction to graph theory, Biblioteka Bolonja, Element, Zagreb 2006. (in Croatian)	4	70
Elezović, N.: Discrete probability, Element, Zagreb, 2008. (in Croatian)	8	70

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description

Course title	Mathematics 4 CE	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student ‘s workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. *Course objectives*

Understanding the concepts of vector space and linear operator and description of their properties. Knowledge and application of numerical methods. Acquiring basic knowledge and skills from Fourier analysis.

1.2. *Course enrolment requirements*

Passed Mathematics 1.

1.3. *Expected course learning outcomes*

Demonstrate the ability to express mathematically and think abstractly in linear algebra.
Demonstrate the ability to solve basic problems and draw conclusions in linear algebra.
Apply linear algebra methods to engineering problems.
Describe errors in numerical methods and know the characteristics of finite arithmetic.
Explain the idea and implement algorithms of numerical methods for solving linear systems and regression problems and least squares method.
Implement methods in a programming language and apply them to engineering problems.
Develop the function in the Fourier series and represent the function in the form of the Fourier integral.
Calculate the Fourier transform of the given function.
Formulate and understand the FFT algorithm.

1.4. *Course content*

Linear algebra: Vector spaces and linear operators. Matrix representation of a linear operator. Eigenvalues and spectrum of the linear operator. Matrix decompositions.
Numerical Mathematics: Errors and Finite Arithmetic. Linear systems. Regression. The method of least squares.
Fourier analysis: Fourier series. Fourier integral and Fourier transform. FFT.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[ ] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. *Student’s obligations*

Course attendance, activity, mid-term exams, tests.

1.8. *Evaluation of student’s work*

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1,5	Oral exam	0,5	Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity, continuous knowledge testing (mid-term exams, quizzes, tests), written exam, oral exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Elezović, N.: Fourier series and integral, Laplace transform, (FER) Biblioteka Bolonja, Element, 2006. (in Croatian)
Elezović N., Aglić A., Linear algebra - a collection of tasks, Element, Zagreb 1999 (in Croatian)
Chapra S. C., Channale R. P.: Numerical methods for engineers, McGrowHill Inc., 1988.

1.11. Optional / additional reading (at the time of proposing study programme)

Kreyszig, E.: Advanced Engineering Mathematics, John Wiley & Sons, Inc., 1993.
Scitovski R.: Numerička matematika, Sveuč. u Osijeku, Elektrotehnički fakultet, 1999.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Elezović, N.: Fourier series and integral, Laplace transform, (FER) Biblioteka Bolonja, Element, 2006. (in Croatian)	*10*	*70*
Elezović N., Aglić A., Linear algebra - a collection of tasks, Element, Zagreb 1999 (in Croatian)	*10*	*70*
Chapra S. C., Channale R. P.: Numerical methods for engineers, McGrowHill Inc., 1988.	*15*	*70*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	**Mobile Application Security**	
Study programme	**Undergraduate University Study of Computing**	
Course status	optional	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	3
	Number of hours (L+E+S)	0+30+15

1. COURSE DESCRIPTION

1.1. Course objectives

Understand the basic principles of mobile application security and the architecture of mobile platforms. Recognize the most common vulnerabilities such as insecure data storage, unencrypted communication, weak authentication mechanisms, and insufficient access control. Adopt secure mobile app development practices, including secure data management, the use of encryption, and user identity verification. The goal is to learn to apply current guidelines for the development and maintenance of secure mobile applications. Develop skills in security testing using tools for mobile app analysis. Learn fundamental techniques to defend against reverse engineering with the aim of protecting source code and sensitive data within the application.

1.2. Course enrolment requirements

1.3. Expected course learning outcomes

Identify the most common vulnerabilities in mobile applications.
Apply tools for testing the security of mobile applications and analyze the level of exposure to security threats.
Apply development practices that address security and effectivenes of techniques to protect against reverse engineering.
Apply current guidelines for developing and maintaining secure mobile applications (e.g. OWASP).

1.4. Course content

Security challenges and threats targeting Android and iOS mobile applications. Implementation of secure development practices, including encryption, authentication, and access control. Use of security testing tools and risk assessment for mobile applications. Identification and analysis of the most common vulnerabilities and methods for their prevention. Introduction to the OWASP Mobile Application Security framework for designing, developing, and testing secure mobile applications. Implementation of techniques to safeguard against reverse engineering and protect mobile applications from malicious modifications.

1.5. Teaching methods	[ ] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Class attendance, complete practice.

1.8. Evaluation of student's work

Course attendance	1.5	Activity/Participation		Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment		Report		Practice	1,5
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Written exam and practice.

1.10. Assigned reading (at the time of the submission of study programme proposal)

OWASP Foundation. OWASP MASTG – Mobile Application Security Testing Guide. Available online at: https://mas.owasp.org/MASTG/

1.11. Optional / additional reading (at the time of proposing study programme)

Velu, Vijay Kumar. Mobile Application Penetration Testing. Birmingham: Packt Publishing, 2017

Levin, Jonathan. macOS and iOS Internals, Volume III: Security & Insecurity. Los Angeles: Technologeeks Publishing, 2016

Elenkov, Nikolay. Android Security Internals: An In-Depth Guide to Android's Security Architecture. San Francisco: No Starch Press

Svensson, Robert. From Hacking to Report Writing: An Introduction to Security and Penetration Testing. Berkeley (CA): Apress, 2016

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
OWASP Foundation. OWASP MASTG – Mobile Application Security Testing Guide. Available online at: https://mas.owasp.org/MASTG/	Free access	-

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	Modelling of Process Information Systems	
Study programme	Undergraduate University Study of Computing	
Course status	elective	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquisition of theoretical knowledge and developing skills for modeling of process information for complex technical systems and electric facilities.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Describe and explain the modular structure of technical systems. Analyze the structure of parameters in electric identification function. Define and distinguish between models of process information in real-time power system. Explain and interpret the sources of process information of technical systems. Describe and correctly interpret the structure of process information in long-distance communication. Design and create UML diagrams for different systems. Distinguish between a standard means of communication and connectivity of open systems. Correctly explain the importance of standardization and application in modeling of process information systems. Define and describe the application of the SCL language. Explain and justify the equipment and software in the control centers of power system.

1.4. Course content

The modular structure of technical systems. Structure of variables in the plant identification function. Display variables in multidimensional vector space. The application of object-oriented approach in modeling. Technological-functional model of process information. Device design of process information. The structure of process information in remote communication between the facilities and control centers. Application of the common data model (CIM). Abstract model of real devices in the facilities. Application program interface management system (EMS-API). Standardization of communication and process information of substation automation. Models of process information in an environment of new technologies and related standards. Open System Interconnection (OSI). Application of UML diagrams for modeling process information. Application of SCL language (based on XML) for configuring and parameterization of intelligent electronic devices (IED). The application of multi-agent system.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [X] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity, preparation of seminar papers, studying.

1.8. Evaluation of student's work

Course attendance	1,5	Activity/Participation		Seminar paper	1,5	Experimental work	
Written exam	1,5	Oral exam	1,5	Essay		Research	
Project		Continuous assessment		Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, activity, seminar paper, written and oral exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Šimunić, J.: Lectures, 2012. (in Croatian)
Shahidehpour M., Wang Y., Communication and Control in Electric Power Systems, Wiley & Sons, 2003..

1.11. Optional / additional reading (at the time of proposing study programme)

Strauss, C.: Practical Electrical Network Automation and Communication Systems, Elsevier, 2003.
Brand, K.P., Lohmann, V., Wimmer, W.: Substation Automation Handbook, UAC, 2003.
Rehtanz, C.: Autonomous systems and intelligent agents in power system control and operation, Springer; 1 ed, 2003.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Šimunić, J.: Lectures, 2012. (in Croatian)	*1*	*14*
Shahidehpour M., Wang Y., Communication and Control in Electric Power Systems, Wiley & Sons, 2003	*1*	*14*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Numerical Methods**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student ‘s workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Identification of numerical problems in engineering. Understanding and use of basic numerical methods. Basic knowledge of Python programming language and corresponding modules. Independent writing of shorter computer programs and use of ready-made modules for preparation, numerical solving and visualization of engineering tasks.

1.2. Course enrolment requirements

Mathematics I.

1.3. Expected course learning outcomes

Suggest a numerical method for solving the mathematical formulation of an engineering problem.
Evaluate the advantages and disadvantages of a particular numerical method.
Compare numerical methods applicable to the same type of problem.
Apply existing computer tools to solve simpler engineering problems.
Build simple computer programs for specific engineering problems using appropriate numerical methods.
Evaluate the results obtained by numerical methods.

1.4. Course content

Solving nonlinear equations: problem formulation and root-finding; overview of numerical methods; convergence and robustness; function as a black box; examples in collision detection, inverse problems based on simulations/software; computer programs in Python.
Optimization methods: 1D bracketing methods; gradient methods; Nelder-mead and pattern search methods; local and global minima; function as a black box; engineering examples; model fitting; computer programs in Python.
Interpolation: Lagrange interpolation; spline curves; Bezier curves and surfaces; B-spline and NURBS; multi-dimensional interpolations; concepts and characteristics of methods; application in data analysis, application in computer graphics; computer programs in Python.
Numerical integration and differentiation: overview of numerical methods; examples on measured/collected data; computer programs in Python.
Solving ordinary differential equations: overview and concepts of numerical methods; local and global errors; examples of solving models (epidemic models, motion, saturation); computer programs in Python.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[ ] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________

1.6. Comments	-

1.7. Student's obligations

Course attendance, mid-term exams, computer knowledge checks.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1	Oral exam	1	Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, mid-term exams, computer knowledge checks, written and/or oral exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Kong Q., Siauw T., Bayen A., Python Programming And Numerical Methods: A Guide For Engineers And Scientists. Elsevier Inc., 2020., Paperback ISBN: 9780128195499, eBook ISBN: 9780128195505
Chapra, Steven C., Canale, Raymond P., Numerical methods for engineers, Eight edition. New York, NY : McGraw-Hill Education, 2021, ISBN 1260232077

1.11. Optional / additional reading (at the time of proposing study programme)

-

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Kong Q., Siauw T., Bayen A., Python Programming And Numerical Methods: A Guide For Engineers And Scientists. Elsevier Inc., 2020., Paperback ISBN: 9780128195499, eBook ISBN: 9780128195505	6	100
Chapra, Steven C., Canale, Raymond P., Numerical methods for engineers, Eight edition. New York, NY : McGraw-Hill Education, 2021, ISBN 1260232077	6	100

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	Operating Systems 1	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquiring fundamental knowledge of modern operating systems.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Understand basic concepts, components and services of operating systems.
Understand the relation between the hardware, software support and functions of the operating system.
Use command-line interface to manage the operating system.
Manage computer processes.
Analyse and implement multi-threaded and multi-process programs.
Manage synchronization mechanisms.
Understand scheduling algorithms.
Manage memory resources.

1.4. Course content

Introduction to operating systems: history of operating systems, operating system structure, interaction between operating system and hardware. Process management: processes and threads, concurrent execution, scheduling, deadlocks, synchronization. Memory management. Shells for working with operating systems and shell programming. Operating system security. Examples of installing and configuring operating systems.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [X] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Class attendance, homework, studying.

1.8. Evaluation of student's work

Course attendance	1	Activity/Participation		Seminar paper	1	Experimental work	

Written exam	2	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Midterm exams (Continuous assessment), seminar paper, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

-

1.11. Optional / additional reading (at the time of proposing study programme)

Silberschatz, Galvin, Gagne: Operating System Concepts, Wiley, 8th Ed.
Budin, Golub, Jakobović, Jelenković: Operating Systems (in Croatian)
Tanenbaum: Modern Operating Systems, Prentice Hall, 2008.
Stallings: Operating Systems: Internals and Design Principles, Prentice Hall, 6th Ed

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	Operating Systems 2	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

This course builds on the topics covered in a basic course on operating systems. Some important topics of modern operating systems such as virtualization, file systems and computer security are covered in detail. The main goal is to gain a solid understanding of these topics to be immediately productive in installing, configuring, managing and using modern operating systems.

1.2. Course enrolment requirements

Enrolled Operating systems 1

1.3. Expected course learning outcomes

Assess the possibilities of virtualization.
Manage virtualization systems.
Describe the types of file systems.
Manage the file subsystem.
Explain basic terms and concepts related to computer security.
Describe the types of security threats and attacks and the most common defence methods.
Describe the properties of frequently used cryptographic primitives.
Explain the role of public key infrastructure and the TLS protocol.
Apply the basic principles of defensive programming.
Describe the basic principles of digital forensics.
Implement a simple attack on a vulnerable system.

1.4. Course content

Virtualization: reasons for virtualization, types of virtualization, virtual machine, usage examples. Modern filesystems: implementation, management and optimization, network filesystems, encrypted filesystems, case-studies of selected filesystems. Computer security: authentication, privacy, public key infrastructure, selected operating system security tools and techniques.

1.5. Teaching methods	[X] lectures [X] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [X] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Class attendance, labs, homeworks, and seminar paper.

1.8. Evaluation of student's work

Course attendance	1	Activity/Participation		Seminar paper	1	Experimental work	
Written exam	2	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Lab exercises, seminar paper, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

1.11. Optional / additional reading (at the time of proposing study programme)

Silberschatz, Galvin, Gagne: Operating System Concepts, Wiley, 7th Ed.
Tanenbaum: Modern Operating Systems, Prentice Hall, 2008.
Stallings: Operating Systems: Internals and Design Principles, Prentice Hall, 6th Ed.
Budin, Golub, Jakobović, Jelenković: Operacijski sustavi

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	Organization of Business Systems	
Study programme	Undergraduate University Study of Computing	
Course status	elective	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Acquiring knowledge in the field of business systems organisation.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Explain the concept of a business system and setting up a business system.
Differentiate between the organizational forms of business systems and specificities.
Explain the organization's resources and the functioning of competitive markets.
Identify risks associated with investments and market participation.
Recognize key factors when making business decisions in the company.
Explain the types and importance of intellectual property.
Distinguish between ownership, management and leadership.
Define job evaluation and teamwork.
Explain organizational behaviour, business policy and business ethics.
Describe the principles and methods of planning.
Explain the influence of supply chains on the success of the business system.
Know the effects of operations and basic financial statements.

1.4. Course content

Definition and evolution of business system organisation. Organisational forms of business systems. Setting up a business system. Basic principles of organisation. Manageability of the system. Formal and informal organisation. Information in the business system. The behavioural approach in organisational theory. Business decision-making. Types of organisational structures. Designing the organisation of the business system. Forms of commercial companies. Supply chains. Organisational changes. Intellectual Property. Organisational behaviour and organisational culture. Business ethics. Job evaluation. Ownership. Management. Leadership. Teamwork. Business policy. Planning. Long-term and short-term business system plans. Economic aspects of business systems. Revenues and expenses. Break-even threshold. Financial statements. Business effects. Organisational resources and competitiveness. Case studies: Study of cases from business practice.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________

1.6. Comments	

1.7. Student's obligations

Attendance, class participation, independent learning.

1.8. Evaluation of student's work

Course attendance	1,5	Activity/Participation		Seminar paper		Experimental work	
Written exam	2	Oral exam		Essay		Research	
Project		Continuous assessment	2,5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Attendance, class activity, continuous assessment, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

T. Mikac, M. Ikonić.: Organizacija poslovnih sustava, Tehnički fakultet Sveučilišta u Rijeci, Rijeka, 2008.

1.11. Optional / additional reading (at the time of proposing study programme)

Sikavica P.; Novak, M.: Poslovna organizacija, informator, Zagreb, 1999.
J. Heizer, B. Render, „Principles of Operations Management", UK/USA: Pearson Publishing, 2013.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Professional Practice 1**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	2	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	-

1. COURSE DESCRIPTION

1.1. Course objectives

Student verifies and complements his own expertise, along with a comprehensive view of the work process.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Apply acquired knowledge and skills from studied courses professional content. Gain working process experience. Develop and further improve competence for solving specific professional engineering problems.

1.4. Course content

Industrial practice within Undergraduate University Study of Naval Architecture is carried out individually in work organization that is engaged in the student's field of study, and with activities in accordance with the Industrial Practice Rules and Study Program curriculum. Within such practice, student is familiarized with the corresponding jobs that are studied through programs of education, with the task of verifying and complementing their own expertise, along with a comprehensive view of the work process.

1.5. Teaching methods

[ ] lectures
[ ] seminars and workshops
[ ] exercises
[ ] long distance education
[X] fieldwork

[ ] individual assignment
[ ] multimedia and network
[ ] laboratories
[ ] mentorship
[ ] other ________________

1.6. Comments

1.7. Student's obligations

Conducting professional practice in duration of 15 working days, or 120 hours, and writing the corresponding report.

1.8. Evaluation of student's work

Course attendance		Activity/Participation		Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project		Continuous assessment		Report	1	Practice	5
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Assesses and evaluates student work and dedication, and written report.

1.10. Assigned reading (at the time of the submission of study programme proposal)

-

1.11. Optional / additional reading (at the time of proposing study programme)

-

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Programming 1**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Introduction to basics of hardware and software. Learning about the process of writing and debugging a program. Introduction to programming principles.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Use basic operating system commands.
Understand the relationship between software and hardware.
Ability to apply a higher-level programming language to solving simple algorithms.
Explain the method of data storage in the computer.
Apply the basic principles of program design.
Choose options using arrays, branching, loops and functions.
Describe the mechanisms of calling functions and assigning parameters.
Design, implement and test simpler programs and find errors.

1.4. Course content

Introduction to computer science. Information coding: number systems, formats and standards for number presentation. Computer architecture: model of a simple processor, instruction execution, process of program compilation. Introduction to operating systems Windows and Linux. Programming principles. Syntax of a programming language. Loops. Arrays. Functions.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments	-	

1.7. Student's obligations

Course attendance, activity, homework, continuous knowledge testing , written exam.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation		Seminar paper		Experimental work	
Written exam	1.5	Oral exam		Essay		Research	

Project		Continuous assessment	2.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Lecturing with knowledge checking based on quizies, partial exams and homeworks. Exercises with problem solving and prepearing for final project.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Darko Grundler: Applied Computing, Graphis Zagreb 2000, ISBN: 953-6647- 03-6. (in Croatian)
Rajko Vulin: From Now we are Programming in C, Turbo C, Školska knjiga, Zagreb 1991. (in Croatian)

1.11. Optional / additional reading (at the time of proposing study programme)

Kernighan B. W., Ritchie D. M., The C Programming Language, Prentice Hall, Inc., 1988.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Darko Grundler: Applied Computing, Graphis Zagreb 2000, ISBN: 953-6647- 03-6. (in Croatian)	1	50
Rajko Vulin: From Now we are Programming in C, Turbo C, Školska knjiga, Zagreb 1991. (in Croatian)	1	50

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Programming 2**	
Study programme	**Undergraduate University Study of Computing**	
Course status	compulsory	
Year	1	
ECTS credits and teaching	ECTS student 's workload coefficient	7
	Number of hours (L+E+S)	45+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Software development in programming language C.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Build a software solution for managing different types of files.
Manage pointers and dynamic memory allocation.
Understand and use advanced syntax constructs in the target programming language.
Propose a programming solution based on complex data types and structures.
Organize program code based on function calls and parameter passing.
Build a software solution based on a standard library of functions.

1.4. Course content

Program structure. Memory classes. Functions and method of passing arguments. Recursive functions. Function libraries. Pointers. Pointer Arithmetic. Pointer and Field Connection. Pointers as function arguments. Dynamic memory allocation. Command line arguments. Structures. Structure fields. Pointers and structures. Self-referential structures. Working with files. Standard functions for reading and writing. Formatted and binary files. Preprocessor. Translation of the program. Configure and make tools. Makefile file. Sorting algorithms. Search algorithms.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [X] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, activity, homework, continuous knowledge testing, written exam.

1.8. Evaluation of student's work

Course attendance	2,5	Activity/Participation		Seminar paper		Experimental work	1
Written exam	1,5	Oral exam		Essay		Research	

Project		Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Lecturing with knowledge checking based on quizies, partial exams and homeworks. Exercises with problems solving and prepearing for final project.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Rajko Vulin: From Now we are Programming in C, Turbo C, Školska knjiga, Zagreb 1991. (in Croatian) Kernighan B. W., Ritchie D. M., The C Programming Language, Prentice Hall, Inc., 1988.

1.11. Optional / additional reading (at the time of proposing study programme)

Rajko Vulin: " A collection of solved tasks from C ", Školska knjiga, Zagreb 1995. (in Croatian)

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Rajko Vulin: From Now we are Programming in C, Turbo C, Školska knjiga, Zagreb 1991. (in Croatian)	*1*	*50*
Rajko Vulin: From Now we are Programming in C, Turbo C, Školska knjiga, Zagreb 1991. (in Croatian)	*1*	*50*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**React and .NET for Full-stack Application Development**	
Study programme	**Undergraduate University Study of Computing**	
Course status	optional	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	3
	Number of hours (L+E+S)	0+45+0

1. COURSE DESCRIPTION

1.1. Course objectives

Master basic and certain advanced concepts of frontend and backend application development using React and .NET technologies, as well as relational database modeling. Understand the way frontend and backend applications communicate.

1.2. Course enrolment requirements

Passed courses Introduction to Object-oriented Programming and Database Systems.

1.3. Expected course learning outcomes

Understand the full-stack application development methodology.
Apply React and .NET technologies to develop full-stack applications.
Independently develop simple and medium complex web applications.

1.4. Course content

JavaScript and TypeScript programming languages. Basics of the C# programming language. Graphical interface development using React. Web API development using ASP.NET Core. Connecting the frontend and backend part of the application. Implementation of authentication, authorization, validation, logging.

1.5. Teaching methods	[ ] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments	-	

1.7. Student's obligations

Attending classes, solving weekly tasks, creating a project.

1.8. Evaluation of student's work

Course attendance	1.5	Activity/Participation		Seminar paper		Experimental work	
Written exam		Oral exam		Essay		Research	
Project	1	Continuous assessment	0.5	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Delivery and presentation of the project.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Andrew Lock, ASP.NET Core in Action, Manning, Third edition, 2023
Robin Wieruch, The Road to React: Your journey to master plain yet pragmatic React.js, Independently published, 2018

1.11. Optional / additional reading (at the time of proposing study programme)

-

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Andrew Lock, ASP.NET Core in Action, Manning, Third edition, 2023	-	-
Robin Wieruch, The Road to React: Your journey to master plain yet pragmatic React.js, Independently published, 2018	-	-

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	**Robotic Systems**	
Study programme	**Undergraduate University Study of Computing**	
Course status	elective	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

Defining the types and characteristics of serial, parallel, and mobile robots used in industrial environments. Understanding the modeling of robot kinematics and dynamics, as well as trajectory planning. Acquiring knowledge about navigation and mapping for mobile robots. Application of numerical simulations of robotic systems. Application of software packages for robotics simulations. Modeling of multi-robot collaborative systems.

1.2. Course enrolment requirements

Passed Mathematics 1 and 2

1.3. Expected course learning outcomes

After successfully completing the course, students will be able to describe the types, characteristics, and components of serial, parallel, and mobile robots. They will have the skills to define and apply algorithms for direct and inverse kinematics of robots, as well as for path planning. Additionally, they will be able to define and apply algorithms for modeling robot dynamics and create simulations of robotic systems using appropriate software packages. Furthermore, students will have the knowledge to define and apply algorithms for simultaneous localization and mapping (SLAM), as well as algorithms for space exploration. Finally, they will be able to define multi-robot systems of various types and their characteristics.

1.4. Course content

Overview of types of robotic manipulators: serial, parallel, and mobile robots. Review of robot kinematics. Dynamics of different types of robots. Denavit-Hartenberg method. Newton-Euler and Lagrange-Euler algorithms. Continuous path planning. Point-to-point path planning. Algorithms for simultaneous localization and manipulation. Application of Python programming language and NumPy library for numerical simulations in robotics. Simulations of robotic systems in existing software packages. Cooperative robotic systems.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [X] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Attendance, participation in activities, completion of programming tasks, self-studying.

1.8. Evaluation of student's work

Course attendance	1	Activity/Participation	1	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project		Continuous assessment	2	Report		Practice	
Portfolio		Laboratories	1				

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Attendance, laboratory exercises, quizzes (3), self-work, presentation of work, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

1. B. Siciliano, K. Oussama: Springer handbook of robotics. Springer, 2016.
2. Kelly, Alonzo. Mobile robotics: mathematics, models, and methods. Cambridge University Press, 2013.
3. Yoshikawa, T. (2010). Foundations of robotics: analysis and control. 3rd ed. MIT press.

1.11. Optional / additional reading (at the time of proposing study programme)

Tsai, Lung-Wen. Robot analysis: the mechanics of serial and parallel manipulators. John Wiley & Sons, 1999.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
B. Siciliano, K. Oussama: Springer handbook of robotics. Springer, 2016.	1	-
Kelly, Alonzo. Mobile robotics: mathematics, models, and methods. Cambridge University Press, 2013.	1	-
Yoshikawa, T. (2010). Foundations of robotics: analysis and control. 3rd ed. MIT press.	1	-

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	Signals and Systems	
Study programme	Undergraduate University Study of Computing	
Course status	elective	
Year	3	
ECTS credits and teaching	ECTS student 's workload coefficient	6
	Number of hours (L+E+S)	45+15+0

1. COURSE DESCRIPTION

1.1. Course objectives

Identify properties of the signals and systems with a goal of their classification. Calculate the response of the linear and time-invariant (LTI) time-continuous and time-discrete systems using the convolutional integral and the convolutional sum in the time domain. Select and apply the appropriate Fourier method for spectral analysis of the periodic and non-periodic signals. Calculate the response of the LTI system in the frequency domain. Explain the relation between the time-continuous and time-discrete signals through the process of signal sampling and reconstruction.

1.2. Course enrolment requirements

Passed Mathematics I and Mathematics II.

1.3. Expected course learning outcomes

Define both elementary signals and basic system properties.
Define the response of LTI systems, convolution integral and sum.
Use the convolution for the time-domain analysis of LTI systems.
Define Fourier series and Fourier transform.
Use different Fourier representations in spectral analysis of signals.
Define the frequency response of LTI systems.
Study LTI systems in the frequency domain. Describe signal sampling and reconstruction procedures.

1.4. Course content

Signals and systems; classification, elementary signals, signal models, operations on signals, system properties. Continuous and discrete LTI systems; zero-input response, zero-state response, convolution of signals, properties of LTI systems. Fourier series; line spectrum, systems with periodic inputs. Fourier transform; signal energy, system frequency response, ideal filters. Signal sampling; aliasing, reconstruction filter. Discrete Fourier Transform (DFT); signal spectral analysis.

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [ ] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student's obligations

Course attendance, project work, individual studying.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation	1	Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Continuous assessment (two written tests), project report, final written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

B. P. Lathi: Linear Systems and Signals, 2/E, Oxford University Press, 2004.
S. S. Soliman and M. D. Srinath: Continuous and Discrete Signals and Systems, 2/E, Prentice Hall, 1998.
M. Vrankić: Signali i sustavi: zbirka riješenih zadataka, Graphis, 2007.

1.11. Optional / additional reading (at the time of proposing study programme)

C. L. Philips, J. Parr, and E. Riskin: Signals, Systems, and Transforms, 4/E, Prentice Hall, 2008.
J. H. McClellan, R. W. Schafer, and M. A. Yoder: Signal Processing First, Prentice Hall, 2003.
S. Haykin and B. Van Veen: Signals and Systems, 2/E, Wiley, 2003.

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
B. P. Lathi: Linear Systems and Signals, 2/E, Oxford University Press, 2004.	*1*	*60*
S. S. Soliman and M. D. Srinath: Continuous and Discrete Signals and Systems, 2/E, Prentice Hall, 1998.	*1*	*60*
M. Vrankić: Signali i sustavi: zbirka riješenih zadataka, Graphis, 2007.	*1*	*60*

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Basic description		
Course title	Software Engineering	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student ‘s workload coefficient	6
	Number of hours (L+E+S)	30+30+15

1. COURSE DESCRIPTION

1.1. Course objectives

Introduction to software engineering basics. Explore lifecycle phases, acquire knowledge and skills of software project management. Teamwork in software development projects.

1.2. Course enrolment requirements

Enrolled Introduction to Object Oriented Programming.

1.3. Expected course learning outcomes

Explain main activities and purpose of software engineering discipline.
Describe software lifecycle phases.
Differentiate software development models and apply agile methodology.
Explain software quality assurance techniques at module level and understand their application at system or organization level. Summarise the project and project results.
Work organization in small teams on the example of a medium-complex project.
Apply the UML language to model different views of the application.
Use appropriate tools for documentation, modeling, design and debugging.
Explain elements and models of software engineering management in software engineering discipline.
Organize a project team in software engineering.
Document and present project progress and results in software engineering.

1.4. Course content

Introduction into software engineering discipline. Software lifecycle model, analysis, specification, design, implementation, and test of requirements. Methods and tools used in each software lifecycle phase. Software development lifecycle models, waterfall, spiral, iterative, incremental, and agile methods. Requirements engineering and software design. Object oriented analysis and design. Software measurements. Quality planning and control. Verification and validation. Management in software engineering discipline.

1.5. Teaching methods	[X] lectures [X] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [X] mentorship [ ] other ________________
1.6. Comments		

1.7. Student’s obligations

Course attendance, activity, seminar and homework, studying, project execution.

1.8. Evaluation of student's work

Course attendance	2	Activity/Participation	0.5	Seminar paper	0.5	Experimental work	
Written exam	1	Oral exam		Essay		Research	
Project	1	Continuous assessment	1	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Course attendance, laboratory exercises, homework, seminar paper, written exam.

1.10. Assigned reading (at the time of the submission of study programme proposal)

Vliet, H.v.: Software Engineering, Principles and Practice. John Wiley & Sons, Chichester, 2009

1.11. Optional / additional reading (at the time of proposing study programme)

Kerzner, H.: Project Management: A Systems Approach to Planning, Scheduling and Controlling, John Wiley & Sons, Hoboken, 2003
I. Sommerville, Software Engineering, 10th Edition, Pearson Education, 2016

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
Vliet, H.v.: Software Engineering, Principles and Practice. John Wiley & Sons, Chichester, 2009	1	59

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.

Sveučilište u Rijeci • University of Rijeka

Trg braće Mažuranića 10 • 51 000 Rijeka • Croatia
T: (051) 406-500 • F: (051) 216-671; 216-091
W: www.uniri.hr • E: ured@uniri.hr

Basic description		
Course title	Web Application Development	
Study programme	Undergraduate University Study of Computing	
Course status	compulsory	
Year	3	
ECTS credits and teaching	ECTS student ‘s workload coefficient	7
	Number of hours (L+E+S)	30+30+0

1. COURSE DESCRIPTION

1.1. Course objectives

This course prepares students to work in the area of web application development by teaching them basics of web systems design and implementation. It is expected to provide practical skills for development of dynamic and interactive web applications by introducing contemporary technologies, platforms, programming languages, and related development tools.

1.2. Course enrolment requirements

None.

1.3. Expected course learning outcomes

Identify client and server web technologies and select the appropriate ones for creating a web application.
Combine various technologies to develop a web application and anticipate potential extensions.
Analyse different approaches' possibilities in constructing web applications.
Apply contemporary technologies for developing both the client and server sides of the web system.
Implement dynamic web applications based on data resources.
Utilize contemporary development environments and frameworks on both the client and server sides.

1.4. Course content

The basic principles for building distributed, dynamic, and interactive information services for content management. Main concepts of the web programming. Design and implementation of web application frontend (HTML, CSS, JavaScript) and backend. Practical examples of dynamic web application development with the use of contemporary technologies. Web services (REST).

1.5. Teaching methods	[X] lectures [ ] seminars and workshops [X] exercises [ ] long distance education [ ] fieldwork	[X] individual assignment [ ] multimedia and network [X] laboratories [ ] mentorship [ ] other ________________
1.6. Comments		

1.7. Student’s obligations

Class attendance, participation in the student project team (group project assignment).

1.8. Evaluation of student’s work

Course attendance	1	Activity/Participation		Seminar paper		Experimental work	
Written exam	1	Oral exam		Essay		Research	

Project	2	Continuous assessment	2	Report		Practice	
Portfolio							

1.9. Procedure and examples of learning outcome assessment in class and at the final exam

Class attendance, midterm exams (continuous knowledge examination), laboratory exercises (individual assignments), and project assignment (participation in a team project).

1.10. Assigned reading (at the time of the submission of study programme proposal)

John Dean (2018.), Web Programming with HTML5, CSS, and JavaScript, Jones & Bartlett Learning
Daniel Correa, Paola Vallejo (2022.), Practical Laravel: Develop clean MVC web applications, Independently
Michael Mikowski, Josh Powell (2013.), Single Page Web Applications, Manning Publications

1.11. Optional / additional reading (at the time of proposing study programme)

1. Andy Budd, Emil Björklund: CSS Mastery, Apress, 2013
2. K. Scott Allen: What Every Web Developer Should Know About HTTP, OdeToCode LLC, 2012

1.12. Number of assigned reading copies with regard to the number of students currently attending the course

Title	*Number of copies*	*Number of students*
John Dean (2018.), Web Programming with HTML5, CSS, and JavaScript, Jones & Bartlett Learning	-	-
Daniel Correa, Paola Vallejo (2022.), Practical Laravel: Develop clean MVC web applications, Independently	-	-
Michael Mikowski, Josh Powell (2013.), Single Page Web Applications, Manning Publications	-	-

1.13. Quality monitoring methods which ensure acquirement of output knowledge, skills and competences

Through the quality assurance system of the Faculty of Engineering.