SVEUČILIŠTE U SPLITU

FACULTY OF ELECTRICAL ENGINEERING, MECHANICAL ENGINEERING AND NAVAL ARCHITECTURE

DETAILED PROPOSAL OF THE STUDY PROGRAMME

UNDERGRADUATE VOCATIONAL STUDY IN COMPUTING

SPLIT, May 2025

1.1. List of mandatory and elective courses

List of courses								
Year of study:1.								
Semester: I.								
STATUS	CODE	COURSE	HOURS IN SEMESTER*					ECTS
			L	S	AE	LE	DE	
	FENP02	Electrical engineering	30	0	15	15	0	6
	* L = lectures, S = seminars, AE = auditory excercise, LE = laboratory excercise, DE = design excercise							

List of courses								
Year of study:1.								
Semester: II.								
STATUS	CODE	COURSE	HOURS IN SEMESTER*					ECTS
			L	S	AE	LE	DE	
Mandatory	FEMY02	Applied mathematics	30	0	30	0	0	5
	FELP03	Programming 2	60	0	30	30	0	10
	* L = lectures, S = seminars, AE = auditory excercise, LE = laboratory excercise, DE = design excercise							

List of courses								
Year of study:2.								
Semester: III.								
STATUS	CODE	COURSE	HOURS IN SEMESTER*					ECTS
			L	S	AE	LE	DE	
	FELP24	Algorithms and data structures	30	0	0	30	0	5
	* L = lectures, S = seminars, AE = auditory excercise, LE = laboratory excercise, DE = design excercise							

List of courses								
Year of study:2.								
Semester: IV.								
STATUS	CODE	COURSE	HOURS IN SEMESTER*					ECTS
			L	S	AE	LE	DE	
	FELP11	Programming in Java	30	0	0	30	0	6
	* L = lectures, S = seminars, AE = auditory excercise, LE = laboratory excercise, DE = design excercise							
List of courses								

Year of study: 3.								
Semester: V.								
STATUS	CODE	COURSE	HOURS IN SEMESTER*					ECTS
			L	S	AE	LE	DE	
Mandatory	FELP25	Software engineering	30	0	0	30	0	5
* L = lectures, S = seminars, AE = auditory excercise, LE = laboratory excercise, DE = design excercise								

1.2. Course description

<table>
<tr><td><b>NAME OF THE COURSE</b></td><td colspan="7"><b>ELECTRICAL ENGINEERING</b></td></tr>
<tr><td><b>Code</b></td><td>FENP02</td><td>Year of study</td><td colspan="5">1.</td></tr>
<tr><td><b>Course teacher</b></td><td>Vicko Dorić, Ph.D., Associate Professor</td><td>Credits (ECTS)</td><td colspan="5">6</td></tr>
<tr><td rowspan="2">Associate teachers</td><td rowspan="2">Ivana Zulim, Ph.D.</td><td rowspan="2">Type of instruction (number of hours)</td><td>L</td><td>S</td><td>AE</td><td>LE</td><td>DE</td></tr>
<tr><td>30</td><td>0</td><td>15</td><td>15</td><td></td></tr>
<tr><td>Status of the course</td><td>Obligatory</td><td>Percentage of application of e-learning</td><td colspan="5">0</td></tr>
<tr><td colspan="8"><b>COURSE DESCRIPTION</b></td></tr>
</table>

Course objectives	Training students for: - understanding and application of basic principles and laws of electrical engineering, - setting up and solving simple electrical circuits, - permanent adoption and deepening of knowledge in the field of electrical engineering.
Course enrolment requirements and entry competences required for the course	None
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	Students will be able to: - define the fundamental phenomena, the quantities and the laws of electrical engineering, - apply fundamental laws of electrical engineering for the calculation of electromagnetic quantities, - apply methods and techniques for solving of linear electrical networks, - formulate simple electrical networks, - analyse simple electrical networks, - calculate quantities of simple magnetic circuits, - measure basic electrical values (current, voltage, resistance).

Course content broken down in detail by weekly class schedule (syllabus)	Course content	L or S hours	AE hours
	Introduction to Electrical Engineering. Brief history of electrical engineering. SI units.	2	0
	Electric charges. Electrostatic field and potential.	2	1
	Electrical capacity, capacitors.	2	1
	Magnetic field. Magnetic field lines. Magnetic flux..	2	1
	Electromagnetic induction.	2	1
	Electric currents. Ohm's Law. Voltage and Current sources.	2	1
	Kirchhoff's lows. Power and energy of DC current.	2	1
	Analysis methods for linear circuits.	2	2
	Time varying currents and voltages. Alternating currents and voltages. AC currents effects.	2	1
	Average and effective value. I-U characteristics within AC circuits.	2	1
	Power and energy of AC current.	2	1
	Fazor representation of the harmonic voltages and currents. AC circuits analysis using complex number representation.	2	1
	Resonance. Simple time domain problems.	2	1
	List of laboratory or design exercises		LE or DE hours
	Introduction to laboratory setup.		2
	Serial, parallel and combined resistors.		2
	Kirchhoff's lows, superposition principle and Thevenin's theorem.		2
	Resistor, capacitor and inductor in AC circuits.		2
	Serial (voltage) resonance.		2
	Power and energy of AC current.		2

	Practical skills exam.	3

Format of instruction	☒ lectures ☐ seminars and workshops ☒ exercises ☐*on line* in entirety ☐ partial e-learning ☐ field work	☐ independent assignments ☒ multimedia ☒ laboratory ☐ work with mentor ☐ (other)
Studentresponsibilities	The presence on lectures in the amount of at least 70 % of the times scheduled. Performed all required laboratory exercises.	

Screening student work *(name the proportion of ECTS credits for eachactivity so that the total number of ECTS credits is equal to the ECTS value of the course)*	Class attendance	2,0	Research		Practical training	
	Experimental work		Report		Individual work	2,7
	Essay		Seminar essay		Laboratory exercises	0,5
	Tests	0,2	Oral exam		Preparation for laboratory exercises	0,5
	Written exam	0,1	Project		(Other)	

Grading and evaluating student work in class and at the final exam	There are two midterms and final exams. The first midterm exam is after 7 weeks of lecturing and the second one is after the next 6 weeks. In the final exams students take tests they didn't pass on the midterm exams. Each midterm test lasts for the 90 min. and consists of 5 theoretical questions and numerical problems. In order to pass the exam, students are required to finish all laboratory exercises and gain at least 50% of total points at each midterm exam or at the final exam. Final grade is determined after the second final test according to the relative grading system. Students which have passed the exam are divided into 4 groups: top 15% of the students get excellent (5) grade, next 35% very good (4) grade, next 35% good (3) grade and last 15% sufficient (2) grade. Students which have failed both final exam, have another exam in the autumn examination periods. Exam lasts for the 90 min. and consists of 5 theoretical questions and numerical problems. Students who gain more than 50% on the last exam are given sufficient (2) grade.

Required literature (available in the library and via other media)	**Title**	**Number of copies in the library**	**Availability via other media**
	V. Pinter: Osnove elektrotehnike, Tehnička knjiga, Zagreb, 1987.	5	
	Felja, I., Koračin, D.: "Zbirka zadataka i riješenih primjera iz osnova elektrotehnike (I i II dio)", Zagreb	5	
	E. Šehović, i drugi: Osnove elektrotehnike zbirka primjera (prvi dio), Školska knjiga, Zagreb, 1992.	5	

Optional literature (at the time of submission of study programme proposal)	B. Jajac: Teorijske osnove elektrotehnike, svezak 1, Graphis, Zagreb, 1998. B. Jajac: Teorijske osnove elektrotehnike, svezak 2, Graphis, Zagreb, 2002.
Quality assurance methods that ensure the acquisition of exit competences	- Evaluation of results in accordance with the above learning outcomes - Feedback from students via surveys - Self-evaluation of teachers - Institutional and non-institutional evaluations
Other (as the proposer wishes to add)	

<table>
<tr><td>NAME OF THE COURSE</td><td colspan="7">APPLIED MATHEMATICS</td></tr>
<tr><td><b>Code</b></td><td>FEMY02</td><td>Year of study</td><td colspan="5">1</td></tr>
<tr><td><b>Course teacher</b></td><td>M.Sc. Ivančica Mirošević</td><td>Credits (ECTS)</td><td colspan="5">5</td></tr>
<tr><td rowspan="2">Associate teachers</td><td rowspan="2">Lea Dujić</td><td rowspan="2">Type of instruction (number of hours)</td><td>L</td><td>S</td><td>AE</td><td>LE</td><td>DE</td></tr>
<tr><td>30</td><td></td><td>30</td><td></td><td></td></tr>
<tr><td>Status of the course</td><td>Obligatory</td><td>Percentage of application of e-learning</td><td colspan="5">10</td></tr>
<tr><td colspan="8"><b>COURSE DESCRIPTION</b></td></tr>
<tr><td>Course objectives</td><td colspan="7">Training students for:
- application of mathematical concepts and tools from the area of ordinary differential equations, numerical mathematics, statistics and probability to analyze and solve engineering problems.</td></tr>
<tr><td>Course enrolment requirements and entry competences required for the course</td><td colspan="7">Good knowledge of High School mathematics and passed State Exam in Mathematics.</td></tr>
<tr><td>Learning outcomes expected at the level of the course (4 to 10 learning outcomes)</td><td colspan="7">Students will be able to:
- state definitions and theorems from the enitre course,
- illustrate theorems with examples,
- solve some first and second order differential equations,
- apply Laplace transform to linear differential equations
- find approximate solution of a nonlinear equation
- approximate function with Lagrange interpolation polynomial
- approximate empirical data with constant, linear or quadratic function
- solve definite integral and Cauchy problem of the first order approximately
- use statistical techniques in data analysis
- find probability distributions of random variables in random experiments</td></tr>
<tr><td rowspan="14">Course content broken down in detail by weekly class schedule (syllabus)</td><td colspan="5">Course content</td><td>L or S hours</td><td>AE hours</td></tr>
<tr><td colspan="5">1. Introduction to Differential Equations. Basic concepts and definitions. Equations with separable variables.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">2. Homogeneous differential equations. Linear differential equations of the first order.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">3. Differential equations of ¸the second order. Linear differential equations of the second order with constant coefficients.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">4. Laplace transform – definition and basic properties. Inverse Laplace transform and basic properties.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">5. Solving linear differential equations with with constant coefficients using Laplace transform.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">6. Introduction to Numerical mathematics. Solving nonlinear equations. Graphical method. Bisection method. Iterative method.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">7. Lagrange interpolation polynomial</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">8. Least square method. Approximating empirical data with constant, linear or quadratic function.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">9. Numerical integration. Trapezoidal rule. Simpson's rule. Euler's method for Cauchy problems.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">10. Descriptive statistics. Discrete data and continuous data. Numerical characteristics.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">11. Introduction to Probability theory. Elementary outcomes. Basics of Combinatorics.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">12. Discrete random variable. Expectation and variance. Binomial distribution. Poisson distribution.</td><td>2</td><td>2</td></tr>
<tr><td colspan="5">13. Continuous random variable. Expectation and variance. Normal distribution.</td><td>2</td><td>2</td></tr>
</table>

	List of laboratory or design exercises	LE or DE hours

Format of instruction	☒lectures ☐seminars and workshops ☒exercises ☐*on line*in entirety ☐partial e-learning ☐field work	☒independent assignments ☐multimedia ☐laboratory ☐work with mentor ☐ (other)
Studentresponsibilities	Regular attendence to and active participation in lectures and excercises.	

Screening student work *(name the proportion of ECTS credits for eachactivity so that the total number of ECTS credits is equal to the ECTS value of the course)*	Class attendance	2	Research		Practical training	
	Experimental work		Report		Self study	2.6
	Essay		Seminar essay		(Other)	
	Tests	0.2	Oral exam		(Other)	
	Written exam	0.2	Project		(Other)	

Grading and evaluating student work in class and at the final exam	During semester two mid-term exams are held. The first exam is scheduled after 7 weeks of lectures, and the second in the week following the lectures. At each mid-term exam students can get 40 points, while the remaining 20 points are attained through assignements during lectures and excercises. The condition for passing the course is minimum 20 points on each mid-term exams and a total of at least 50 points. After semester, two final exams and a correction exam are held. Students which did not pass one mid-term exam, can take only this part of the exam during final exams. Students which did not pass any mid-term exam, take the final exam with comprehensive course content. In that case, maximum numbers of available points is 80. The condition for passing the course is minimum 40 points in the final exam and a total of at least 50 points. The grade is formed after the second final exam according to article 75 of the Statute of FESB: 15% of the best students get the mark excellent (5), next 35% students get the mark very good (4), next 35% students get the mark good (3), and the last 15% students get thet mark sufficient (2). Students who did not pass the course after final exams, and have obtained total of at least 10 points, can attend the correction exam. On the correction exam maximal number of points is 100, and the minimum requirement for a passing grade is 50 points. Mid-term exams, final exams and correction exams are held according to the exam schedule.

Required literature (available in the library and via other media)	**Title**	**Number of copies in the library**	**Availability via other media**
	Lecture materials on FESB e-learning portal.		https://elearning.fesb.hr/

Optional literature (at the time of submission of study programme proposal)	T. Bradić, J. Pečarić, R. Roki, M. Strunje: Matematika za tehnološke fakultete, Element, Zagreb, 1998. B. P. Demidovič: Zbirka zadataka iz više matematike, Školska knjiga, Zagreb 1998.

	Ivo Pavlić, Statisticka teorija i primjena, Zagreb, 1971
Quality assurance methods that ensure the acquisition of exit competences	- homework - short tests - quizzes - mid-term exams - final exam - student questionnaires
Other (as the proposer wishes to add)	

<table>
<tr><td>**NAME OF THE COURSE**</td><td colspan="7">**PROGRAMMING 2**</td></tr>
<tr><td>**Code**</td><td>FELP03</td><td>Year of study</td><td colspan="5">1</td></tr>
<tr><td>**Course teacher**</td><td>Linda Vicković, Ph.D., Associate Professor</td><td>Credits (ECTS)</td><td colspan="5">10</td></tr>
<tr><td rowspan="2">Associate teachers</td><td rowspan="2">Ivica Crnjac, Teaching Assistant</td><td rowspan="2">Type of instruction (number of hours)</td><td>L</td><td>S</td><td>AE</td><td>LE</td><td>DE</td></tr>
<tr><td>60</td><td></td><td>30</td><td>30</td><td></td></tr>
<tr><td>Status of the course</td><td>Obligatory</td><td>Percentage of application of e-learning</td><td colspan="5">0</td></tr>
<tr><td colspan="8">**COURSE DESCRIPTION**</td></tr>
<tr><td>Course objectives</td><td colspan="7">Training students for:
- understanding and appliance of basic programming knowledge in C programming language,
- usage of standard functions from C libraries like input / output and mathematical functions,
- Writing C program functions, pointer usage, dynamic memory allocation and structures.</td></tr>
<tr><td>Course enrolment requirements and entry competences required for the course</td><td colspan="7">None</td></tr>
<tr><td>Learning outcomes expected at the level of the course (4 to 10 learning outcomes)</td><td colspan="7">Students will be able to:
- describe fundamentals related to writing, compiling, linking and executing C programs,
- write, build and execute simple C programme,
- using functions, pointers and dynamic memory allocation in programmes,
- using user's data types like structures and unions,
- imply data input from data files and data storage in data files,
- using debugger for problems solving.</td></tr>
<tr><td rowspan="4">Course content broken down in detail by weekly class schedule (syllabus)</td><td colspan="5">Course content</td><td>L or S hours</td><td>AE hours</td></tr>
<tr><td colspan="5">Introduction to C semantic. Comments. Basic data types. Variables.</td><td>4</td><td>2</td></tr>
<tr><td colspan="5">Pre-processor's statements. Arithmetic expressions. Prefix/postfix increment/decrement operators.</td><td>4</td><td>2</td></tr>
<tr><td colspan="5">Data input from keypad. Relation operators. for loop.</td><td>4</td><td>2</td></tr>
</table>

	Making decisions – if statement. Logical operators in compound relations. while loop, do while loop and switch statement.	4	2
	Working with arrays. Defining an array. Character arrays – strings. Standard functions for manipulating arryas of characters. String and char input from keypad.	4	2
	Multidimensional arrays.	4	2
	Functions. Scope of the variable. Parameters transfer by value and by reference. Array as a function's argument. recursive functions	4	2
	Data conversion in C. ASCCI values	4	2
	Structures. Enumerated data type. Unions. Array of structures. Structure containing structures.	4	2
	Pointers. Address operator. Pointer to integer and character. Pointer to arrays of integers and characters. Pointers to structures. Pointers inside structures..	4	2
	Input and output operations with files.	4	2
	Dynamic memory allocation.	4	2
	break, continue statements. exit function. System calls. Arguments of the main function. Pre-processors statements. Conditional compilation, Pointers to functions.	4	2
	List of laboratory or design exercises		LE or DE hours
	First C program. Program compiling, linking and executing. Writing to the screen. For loop examples		2
	Data input from keypad. If statement and logical operators in compound relations.		2
	while loop, do-while loop and random numbers.		2
	Switch statement and integer arrays.		2
	Character arrays and standard functions for manipulating character arrays.		2
	Two-dimensional arrays of integers.		2
	Functions		2
	Recursive functions		2
	Structures.		2
	Pointers to basic data types. Pointers to arrays and structures.		2
	Input and output operations with files.		2
	Dynamic memory allocation.		2
Format of instruction	☒ lectures ☐ seminars and workshops ☒ exercises ☐*on line* in entirety ☐ partial e-learning ☐ field work	☐ independent assignments ☒ multimedia ☒ laboratory ☐ work with mentor ☐ (other)	
Studentresponsibilities	The presence on lectures in the amount of at least 70 % of the times scheduled. Performed all required laboratory exercises.		

Screening student work *(name the proportion of ECTS credits for eachactivity so that the total number of ECTS credits is equal to the ECTS value of the course)*	Class attendance	4	Research		Practical training	
	Experimental work		Report		(Other)	3
	Essay		Seminar essay		(Other)	1,4
	Tests	0,2	Oral exam		(Other)	1,3
	Written exam	0,1	Project		(Other)	

Grading and evaluating student work in class and at the final exam	There are two parts of the exam, theoretical and laboratory part. Laboratory part of exam is held on computers at the end of all laboratory exercises, and after that on final exams. Theoretical part of exam is written and there are two midterms and final exams. The first midterm exam is after 7 weeks of lecturing and the second one is

	after the next 6 weeks. Each midterm test consists of 15 questions some practical and some theoretical. The requirement for passing grade is the positive grade of laboratory part of exam and 50 % points on each midterm exam or the final exam. Grade (in percentage) is formed according to the formula: Grade = 0,5 LV + 0,5 T where: • LV – grade from laboratory part of exam, • T – grade from the theoretical part of exam.		
Required literature (available in the library and via other media)	**Title**	**Number of copies in the library**	**Availability via other media**
	Vicković, L. Programiranje 2, prezentacije s predavanja.		e-learning portal
	Mateljan I. Računala i programiranje, skripta, FESB, Split, 2004		
	Byron S.Gottfried: "Programming with C", Schaum's Outlines, McGraw-Hill, New York, 1996.		
	Besplatne knjige i tečajevi na internetu: http://www.freeprogrammingresources.com/ctutor.html		
Optional literature (at the time of submission of study programme proposal)	-		
Quality assurance methods that ensure the acquisition of exit competences	- Evaluation of results in accordance with the above learning outcomes - Feedback from students via surveys - Self-evaluation of teachers - Institutional and non-institutional evaluations		
Other (as the proposer wishes to add)			

NAME OF THE COURSE	**ALGORITHMS AND DATA STRUCTURES**						
Code	FELP24	Year of study	2.				
Course teacher	Linda Vicković, Ph.D., Associate Professor	Credits (ECTS)	5				
Associate teachers	Ivica Crnjac, Teaching Assistant	Type of instruction (number of hours)	L	S	AE	LE	DE
			30	0	0	30	
Status of the course	Obligatory	Percentage of application of e-learning	0				
COURSE DESCRIPTION							
Course objectives	Training students for: - understanding and appliance of basic algorithm analysis principles, - permanent adoption and deepening of knowledge form the area of dynamic memory allocation, as well as management of abstract data types like stacks, queues and binary trees,						

	- understanding and appliance of simple and complex sorting algorithms.		
Course enrolment requirements and entry competences required for the course	Students have to pass Programming 1 from the first year of study.		
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	Students will be able to: - define basic terms related to algorithm analysis, - describe and perform adding, deleting, searching, of elements in single and double linked lists, - create functions for adding and deleting of stack and queue elements, - recognise appliance of abstract data types in problem solving, - describe steps of adding, deleting and searching of elements in binary search trees, - using basic AVL rotations to reach a balance condition, - name and use different recursive searching algorithms.		
Course content broken down in detail by weekly class schedule (syllabus)	Course content	L or S hours	AE hours
	Introduction to the course. Review of basic elements of C programming language (recursive functions, data structures, pointers, dynamic memory allocation, file handling).	2	
	Algorithm analyses mathematical background and running time calculation of algorithm.	2	
	Abstract data types, simple implementation of linked lists and its basic operations.	2	
	Linked lists sorting.	2	
	Doubly linked lists, circularly linked lists.	2	
	Stack and its applications (stack frames, balancing symbols), queue.	2	
	Binary search trees and basic operations on binary search trees.	2	
	AVL trees.	2	
	Basic sorting methods.	2	
	Shellsort i Quicksort.	2	
	Mergesort.	2	
	Heaps and Heapsort.	2	
	Hashing.	2	
	List of laboratory or design exercises		LE or DE hours
	Basic operations in the array of structures.		2
	Adding new element at the end and beginning of linked list as well as Printing and deleting elements.		2
	Adding new element behind and in front of the specified element in linked list. Sorting of elements in list, reading list elements from file and writing list elements in file.		2
	Using linked lists for polynomial adding and multiplying.		2
	Union and cross section of two linked lists.		2
	Stack and queue implementation of linked lists.		2
	Circular stack and priority queue implementation of linked lists.		2
	Using stack for postfix expression.		2
	Using simple sorting algorithms like exchange, selection, insertion and bubble sort for randomly generated numbers sorting.		2
	Using Shllsort, Quicksort and Mergesort for randomly generated numbers sorting.		2
Format of instruction	☒ lectures ☐ seminars and workshops ☐ exercises ☐*on line* in entirety ☐ partial e-learning ☐ field work	☐ independent assignments ☒ multimedia ☒ laboratory ☐ work with mentor ☐ (other)	

Studentresponsibilities	The presence on lectures in the amount of at least 70 % of the times scheduled. Performed all required laboratory exercises.

Screening student work *(name the proportion of ECTS credits for eachactivity so that the total number of ECTS credits is equal to the ECTS value of the course)*	Class attendance	1	Research		Practical training	
	Experimental work		Report		Individual work	1,5
	Essay		Seminar essay		Laboratory exercises	1,5
	Tests	0,2	Oral exam		Preparation for laboratory exercises	0,7
	Written exam	0,1	Project		(Other)	

Grading and evaluating student work in class and at the final exam	There are two parts of the exam, theoretical and laboratory part. Laboratory part of exam is held on computers at the end of all laboratory exercises, and after that on final exams. Theoretical part of exam is written and there are two midterms and final exams. The first midterm exam is after 7 weeks of lecturing and the second one is after the next 6 weeks. Each midterm test consists of 5 questions some practical and some theoretical. The requirement for passing grade is the positive grade of laboratory part of exam and 50 % points on each midterm exam or the final exam. Grade (in percentage) is formed according to the formula: Grade = 0,5 LV + 0,5 T where: • LV – grade from laboratory part of exam, • T – grade from the theoretical part of exam.

Required literature (available in the library and via other media)	**Title**	**Number of copies in the library**	**Availability via other media**
	• Vicković, L. Algoritms and data structures, lecture notes.		e-learning portal
	• Weiss, M., Data Structures and Algorithm Analysis in C (sections 1-6), Addison-Wesley, 1997.		
	• Sedgewick, R. Algorithms in C, Addison-Wesley, 1990.		

Optional literature (at the time of submission of study programme proposal)	- Neapolitan, R., Naimipour, K. Foundations of Algorithms, Jones & Barlett Learning, 2015.
Quality assurance methods that ensure the acquisition of exit competences	- Evaluation of results in accordance with the above learning outcomes - Feedback from students via surveys - Self-evaluation of teachers - Institutional and non-institutional evaluations
Other (as the proposer wishes to add)	

NAME OF THE COURSE	**PROGRAMMING IN JAVA**						
Code	FELP11	Year of study	3.				
Course teacher	Eugen Mudnić, Ph.D., Assistant Professor	Credits (ECTS)	6				
Associate teachers		Type of instruction (number of hours)	L	S	AE	LE	DE
			30	0	0	30	
Status of the course	Obligatory	Percentage of application of e-learning	0				

COURSE DESCRIPTION

Course objectives	Training students for - Use Java language and environment. - Use object oriented program design.
Course enrolment requirements and entry competences required for the course	Previously taken courses : C programming
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	Students will be able to: - Establish Java development environment. - Write Java applications. - Use object oriented programming model. - Use Java system libraries. - Use complex development environment. - Predict Java code performance.

Course content broken down in detail by weekly class schedule (syllabus)	Course content	L hours	AE hours
	Introduction to Java and comparison to other C languages. Basic Java application.	2	0
	Java class, methods and attributes declaration. Class member access.	2	0
	Encapsulation. Constructors. Packages.	2	0
	Identificators, keywords and data types. Variables, declaration, assignment. Construction and initialization of objects. References. Java coding conventions.	2	0
	Variable scope. Operators. Program flow control (loops and branches). Arrays.	2	0
	Inheritance. Class derivation. Polymorphism. Access control.	2	0
	Methods and constructor overload. Methods override. Object class. Wrapper classes.	2	0
	First midterm exam.		
	Advanced class features. Abstract classes. Interfaces.	2	0
	Exceptions. Exceptions handling. Exception categories. Custom exceptions.	2	0
	Java console applications. Java command line arguments. Using console I/O functions. Using file I/O functions.	2	0
	Java utility classes.	2	0
	Java GUI. Frame and panel components.	2	0
	Java threads. Java threads control. Java threads synchronization.	2	0
	Second midterm exam		
	List of laboratory exercises		LE hours
	Java virtual machine. Hello World application.		2
	Eclipse development environment.		2
	Numbers and Strings. Reading input.		2
	Class design. Class Student.		2
	Java applets.		2
	Conditional operators.		2

	Class definition – class Robot	2
	Arrays and complex data structures.	2
	Class extension. Combining related classes.	2
	Exceptions in input/output operations.	2
	Java threads. Thread management. Thread synchronization.	2
	Java GUI. Event handling.	2
	Java database connection.	2

Format of instruction	☒ lectures ☐ seminars and workshops ☐ exercises ☐*on line* in entirety ☐ partial e-learning ☐ field work	☒ independent assignments ☒ multimedia ☒ laboratory ☐ work with mentor ☐ (other)
Studentresponsibilities	The presence on lectures in the amount of at least 70 % of the times scheduled. Performed all required laboratory exercises.	

Screening student work *(name the proportion of ECTS credits for eachactivity so that the total number of ECTS credits is equal to the ECTS value of the course)*	Class attendance	2,0	Research		Practical training	
	Experimental work		Report		Individual work	2,0
	Essay		Seminar essay	0,2	Laboratory exercises	1,5
	Tests	0,2	Oral exam		Preparation for laboratory exercises	0,0
	Written exam	0,1	Project		(Other)	

Grading and evaluating student work in class and at the final exam	There are two midterms and final exams. The first midterm exam is after 7 weeks of lecturing and the second one is after the next 6 weeks. Each midterm test consists of 20 questions and final tests consist of 20 theoretical questions and numerical problems. In the final exams students that did not pass the midterm exams take part. The midterm and final exams are carried out as written tests. The requirement for passing grade is the positive assessment of laboratory exercises and 50 % points on each midterm exam or the final exam. Grade (in percentage) is formed according to the formula: Grade(%) = 0,05 NP + 0,15 LV + 0,4 (M1 + M2) the activities in percentage: • NP - attendance at lectures, • LV – laboratory assessment, • M1, M2 – test results.

Required literature (available in the library and via other media)	**Title**	**Number of copies in the library**	**Availability via other media**
	E. Mudnic, Authorized lectures.		
	The Java Language Specification, Java SE 7 Edition (Java Series)	0	free available on Internet

Optional literature (at the time of submission of study programme proposal)	The Java Tutorial: A Short Course on the Basics (5th Edition)
Quality assurance methods that ensure the acquisition of exit competences	- Evaluation of results in accordance with the above learning outcomes - Feedback from students via surveys - Self-evaluation of teachers - Institutional and non-institutional evaluations - Feedback from graduated students
Other (as the proposer wishes to add)	

NAME OF THE COURSE	**SOFTWARE ENGINEERING**						
Code	FELP25	Year of study	2.				
Course teacher	Linda Vicković, Ph.D., Associate Professor	Credits (ECTS)	5				
Associate teachers		Type of instruction (number of hours)	L	S	AE	LE	DE
			30	0	0	30	
Status of the course	Obligatory	Percentage of application of e-learning					

COURSE DESCRIPTION

Course objectives	Training students for: - understanding and usage of engineering approach to software development, - how to write user requirements specification, software design specification and test plan documents in software development process, - applying acquired knowledge in the practical software development.
Course enrolment requirements and entry competences required for the course	Students have to pass Object oriented programming from the second year of study.
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	Students will be able to: - define fundamental terms of engineering approach in software development, - identify different steps in software development, - differ agile and classical software development methods, - provide required documents during software development process, - using UML diagrams for software architecture description, - recognize different architecture and design patterns, - describe different software verification and validation phases, - define importance of software evolution.

Course content broken down in detail by weekly class schedule (syllabus)	Course content	L hours	AE hours
	Introduction in Software engineering.	2	0
	Software processes and software process models.	2	0
	Agile software development. Extreme programming..	2	0
	Scrum and Scaling agile methods.	2	0
	Software requirements.	2	0
	The software requirements document. Requirements elicitation, analysis and validation.	2	0
	System modelling. Introduction to UML.	2	0
	Architectural design.	2	0
	Architectural patterns.	2	0
	Design and implementation. Design patterns.	2	0
	Software testing.	2	0
	Test driven development	2	0
	Software maintenance and evolution.	2	0
	List of laboratory or design exercises		LE hours
	Advanced features of Microsoft Office for document formatting.		2
	Using Microsoft Project in project management.		2
	Using Microsoft Visio for system modelling (UML diagrams).		2
	Using testing package in Microsoft Visual Studio.		2
	Visiting lecture – Project management.		2
	Visiting lecture – Estimation effort for software development product.		2
	Visiting lecture – Scrum methodology for software development.		2
	Visiting lecture – Kanban methodology for software development.		2
	Visiting lecture – Software testing		2

	Visiting lecture – Software engineering in Ericsson Nikola Tesla – environment, market and evolution.	2
Format of instruction	☒ lectures ☐ seminars and workshops ☐ exercises ☐*on line* in entirety ☐ partial e-learning ☐ field work	☒ independent assignments ☒ multimedia ☒ laboratory ☐ work with mentor ☐ (other)
Studentresponsibilities	The presence on lectures in the amount of at least 70 % of the times scheduled. Performed all required laboratory exercises.	

Screening student work *(name the proportion of ECTS credits for eachactivity so that the total number of ECTS credits is equal to the ECTS value of the course)*	Class attendance	1	Research		Practical training	1
	Experimental work		Report		Individual work	2
	Essay		Seminar essay		Laboratory exercises	0,5
	Tests	0,2	Oral exam		Preparation for laboratory exercises	0,2
	Written exam	0,1	Project		(Other)	

Grading and evaluating student work in class and at the final exam	There are two parts of the exam, practical and theoretical. For practical part students have to make a software project and related documentations. It is done in groups from 3 to 5 students. Project is divided in three phases and each is graded. Finale project grade is counted as average. Theoretical part of exam is written and there are two midterms and final exams. The first midterm exam is after 7 weeks of lecturing and the second one is after the next 6 weeks. Each midterm test consists of 10 theoretical questions. The requirement for passing grade is the positive grade from project part and 50 % points on each midterm exam or the final exam. Grade (in percentage) is formed according to the formula: $Grade = 0{,}6\ P + 0{,}4\ T$ where: • P – project grade, • T – grade from the theoretical part of exam.

Required literature (available in the library and via other media)	**Title**	**Number of copies in the library**	**Availability via other media**
	Vicković, L. Programsko inženjerstvo, prezentacije s predavanja.		e-learning portal
	Somerville, I. Software engineering, Addison Wesley, 9 edition, 2011.		
	Sach, S. Object Oriented Software Engineering, McGraw-HIll, 2008.		
	Fowler, M. UML Distilled, Addison Wesley, third edition, 2003.		

Optional literature (at the time of submission of study programme proposal)	
Quality assurance methods that ensure the acquisition of exit competences	- Evaluation of results in accordance with the above learning outcomes - Feedback from students via surveys - Self-evaluation of teachers - Institutional and non-institutional evaluations
Other (as the proposer wishes to add	