



POLYTECHNIC OF MEĐIMURJE IN ČAKOVEC

COURSE SYLLABUS

ACADEMIC YEAR: 2020/2021

1. GENERAL COURSE INFORMATION

1.1 Course name	Programming			
1.2 Study program/s	Undergraduate professional study Sustainable Development			
1.3 Course status (O,E)	Obligatory	1.6 Mode of instruction (number of hours)	Lectures	30
1.4 Course code			Exercises	45
1.5 Course abbreviation	PROG		Seminars	
1.6 Semester	2		E-learning	
1.7 ECTS	7	1.7 Place and time of instruction	The premises of Polytechnic of Međimurje in Čakovec, according to schedule published on web pages	

2. TEACHING STAFF

2.1 Course leader/s-title	Bruno Trstenjak, PhD, Senior lecturer, Sanja Brekalo, PhD, High School Professor	contact	btrstenjak@mev.hr sbrekalo@mev.hr
		contact	
2.2 Assistant/s- title		contact	
		contact	
2.3 Instruction held by- title		contact	

3. COURSE DESCRIPTION

3.1 Course goals	After completing the course, the student will be able to apply a structured procedural programming language for solving simple programming problems and create a simple application using computer algorithms. Knowledge in the field of computer programming is acquired, the student enables to independently perform program tasks.
3.2 Prerequisites	None.
3.3 Course outcomes	After successfully completing the course, students will be able to: O1 - Define the variables and data types depending on the terms of reference O2 - Use the basic commands of the program structure depending on the requirements of the program tasks O3 - Apply various data structures in applications O4 - Use functional structures in the development of more complex applications and independently develop a software algorithm for solving problem tasks O5 - Create simple applications or functions based on the application of procedural programming language
3.4 Course content	The course presents content related to the development of desktop applications using a procedural language such as C++ or Java. Emphasis on the application logic program structure, different data structures and the development of algorithms for solving simple programming tasks. In teaching units and exercises, the contents of the lectures are supported by implemented algorithms and solved tasks, both in lectures and in exercises.

3.5 Types of coursework	X	Lectures	X	Exercises		Blended e-learning	X	Individual activities		Laboratory																																				
		Seminars and workshops	X	Distant learning		Field classes		Multimedia and network		Mentorship																																				
		Other																																												
3.6 Language of instruction	Croatian/English																																													
3.7 Monitoring students' work (enter the number of ECTS credits for each activity so that the total number of ECTS credits is equal to the total ECTS value of the course, 1 ECTS = 30 hours)	2,50	Class attendance		0,5	Seminars			Essay																																						
		Class activity			Project			Report/paper																																						
	3,0	Midterm exams			Practical task			Continuous knowledge check																																						
		Written exam			Experimental work																																									
	1,0	Oral exam			Research																																									
3.8 Assessment and evaluation of students' work during classes and at the final exam	<table border="1"> <thead> <tr> <th>Activity specification</th> <th>Percent %</th> <th>Points</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="text-align: center;">Assessment during instruction</td> </tr> <tr> <td>Attendance</td> <td>5%</td> <td>5</td> </tr> <tr> <td>Continuous assessment</td> <td>20%</td> <td>20</td> </tr> <tr> <td>Seminar/ project/ essay</td> <td>5%</td> <td>5</td> </tr> <tr> <td>Midterm exam 1</td> <td>15%</td> <td>15</td> </tr> <tr> <td>Midterm exam 2</td> <td>15%</td> <td>15</td> </tr> <tr> <td>Midterm exam 3</td> <td>15%</td> <td>15</td> </tr> <tr> <td>Oral exam</td> <td>25%</td> <td>25</td> </tr> <tr> <td colspan="3" style="text-align: center;"><i>Exam assessment for the students who failed to fulfill all the obligatory requirements during the semester</i></td> </tr> <tr> <td>Written exam</td> <td>45%</td> <td>45</td> </tr> <tr> <td>Total:</td> <td>100%</td> <td>100</td> </tr> </tbody> </table>										Activity specification	Percent %	Points	Assessment during instruction			Attendance	5%	5	Continuous assessment	20%	20	Seminar/ project/ essay	5%	5	Midterm exam 1	15%	15	Midterm exam 2	15%	15	Midterm exam 3	15%	15	Oral exam	25%	25	<i>Exam assessment for the students who failed to fulfill all the obligatory requirements during the semester</i>			Written exam	45%	45	Total:	100%	100
	Activity specification	Percent %	Points																																											
	Assessment during instruction																																													
	Attendance	5%	5																																											
	Continuous assessment	20%	20																																											
	Seminar/ project/ essay	5%	5																																											
	Midterm exam 1	15%	15																																											
	Midterm exam 2	15%	15																																											
	Midterm exam 3	15%	15																																											
	Oral exam	25%	25																																											
<i>Exam assessment for the students who failed to fulfill all the obligatory requirements during the semester</i>																																														
Written exam	45%	45																																												
Total:	100%	100																																												
3.9 Assessment criteria – analysis per learning outcomes	Ways of evaluating learning outcomes																																													
		Attenda nce	Conti nuous asse.	Mid- term exam 1	Mid- term exam 2	Mid- term exam 3	Sem inar	Oral exam	Total																																					
	Outcome 1			5				5	10																																					
	Outcome 2		5	10	5			5	25																																					
	Outcome 3		5		10			5	20																																					
	Outcome 4		5			5		5	15																																					
	Outcome 5		5			10		5	20																																					
	Outcome not-related	5					5		10																																					
	Total	5	20	15	15	15	5	25	100																																					
	<p>Grading of outcomes (in order to pass the mid-term exam/exam the student must achieve at least 50% points for each learning outcome)</p> <p>Points Grade</p> <p>89 – 100 excellent (5)</p> <p>76 – 88 very good (4)</p> <p>63 – 75 good (3)</p> <p>50 – 62 pass (2)</p> <p>0 – 49 fail (1)</p>																																													

3.10 Specific features related with taking the course	If a student collects 50% of the points of each outcome, he / she directly takes the exam, if he / she has submitted a seminar paper. A student cannot access the exam if he / she has not submitted a seminar paper. Seminar papers are prepared according to the instructions published on the Merlin system and are submitted by posting on the Merlin. The seminar paper should be submitted at least 3 days before the exam deadline.	
3.11 Students obligations	Full-time students are required to attend at least 70% of the total number of hours of lectures and exercises in order to exercise the right to take the exam. Part-time students are required to attend at least 30% of the total number of hours of lectures and exercises in order to exercise the right to take the exam. If the student has not fulfilled all the obligations set by the course, he is obliged to attend the lectures again and meet the conditions for taking the exam. Attendance can be offset by online tuition, organised webinars and added assignments given by teachers. One lesson lasts 45 minutes, and several hours form a teaching unit. Absence from one teaching unit is counted as one absence. Delays and apologies are recorded separately. In that case, if the student missed more than 50% of classes, and has a justifiable reason/apology, the request should be submitted to the Department Council, which then decides on the justification of student absences with the obligatory opinion of the course leader.	
3.12 Written assignments		
3.13 Required reading	1.	Bruno Trstenjak, Programiranje C++, 2020
	2.	
3.14 Additional reading	1.	Julijan Štribar, Boris Motnik: Demistificirani C++, 2018
	2.	Liam Damien : C++: Step by step Beginners Guide in Mastering C++, Independently published, 2019
4 ADDITIONAL COURSE INFORMATION		
4.1 Quality control	The quality of the program, teaching process, teaching skills and level of mastery of the material will be established by conducting a written evaluation based on questionnaires, and in other standardised ways and in accordance with the by-laws of the Polytechnic of Međimurje in Čakovec.	
4.2 Contact the teacher	Students can contact the teacher during the office hours and during classes, while for short questions and explanations they can contact him/her any day during working hours by coming in person or by landline. It is also possible to ask questions by e-mail, which will be answered in 48 hours at the latest. It is desirable for students to come as often as possible for any possible questions during the teacher's office hours.	
4.3 Information about the course	It is the obligation of each student to be regularly informed about the course. All notifications about the classes or possible postponement of classes will be posted on the bulletin board and on the website of the Polytechnic at least 24 hours in advance.	

4.4 Course contribution to the study program	<ul style="list-style-type: none"> - Develop programming code in multiple programming languages using modern methods and tools - Choose ways of structuring data in program code, as well as techniques for writing complex program forms and use standard algorithms
---	---

5. ANALYSIS OF COURSE TOPICS (the number of hours is equal to the number of lectures and exercises of the course)

LECTURES

Hours	Topic and description	Method	Learning outcomes	Course outcome
		<ul style="list-style-type: none"> • Direct teaching (lecture, instruction, pp presentation) • Discovery learning (individual, lead, discussion) • Group learning • Case study • Field classes... 		
1. & 2.	Introduction to the course content, teaching methods, evaluation of student work during the semester Basic concepts of programming, types of errors, types of programming languages, properties of C ++ languages.	Lecture, discussion, PP presentation	Distinguish ways of program development, types of programming languages, types of errors in the process of program development.	O1
3. & 4.	Programming stages, debugger concept, compiler, algorithm concept, program flow diagram.	Lecture, discussion, PP presentation	Distinguish stages in program development. Explain the structure of flowcharts and algorithm operation.	O1
5. & 6.	Fundamentals of C ++ language: identifiers, variables, constants, data types, input / output operations.	Lecture, discussion, PP presentation	Distinguish data types. Apply basic command syntax to define variables and constants.	O2
7. & 8.	Operators: arithmetic, relational, logical, increase / decrease operators, conditional. Analysis of simple examples of solved tasks.	Lecture, discussion, PP presentation	Distinguish types of operators. Apply operators in solving simple tasks.	O2
9. & 10.	Program flow control commands: if, if-else, switch commands.	Lecture, discussion, PP presentation	Distinguish branching commands. Apply commands in solving simple tasks.	O2
11.&12.	Midterm exam 1	Individual work	Outcome check O1, O2	

13.&14.	Loop control instructions (for, while, do-while). Analysis of simple examples of solved tasks.	Lecture, discussion, PP presentation	Distinguish between the loop control instructions. Apply loop instructions in solving simple tasks.	O2
15.&16.	Working with strings (char, string). Basic functions for working with data. Analysis of simple examples of solved tasks.	Lecture, discussion, PP presentation	Explain the properties of character data types. Apply functions to work with characters.	O3
17.&18.	Working with arrays (1D, 2D, 3D array). Analysis of simple examples of solved tasks.	Lecture, discussion, PP presentation	Explain the properties of an array. Apply array structure in solving program tasks.	O3
19.&20.	Midterm exam 2	Individual work	Outcome check O2, O3	
21.&22.	Functions in programming. Basic characteristics of functions, declaration of functions, transfer of arguments, inline functions).	Lecture, discussion, PP presentation	Explain the concept of function and attribute. Apply functions in structured program development.	O4
23.&24.	Vector - basic properties, standard functions in the Vector class. Analysis of simple examples of solved tasks.	Lecture, discussion, PP presentation	Explain the properties of vectors. Apply functions to work with the vector.	O4
25.&26.	File - basic operations for using text files. Analysis of simple examples of solved tasks.	Lecture, discussion, PP presentation	Explain file types and data access functions.	O5
27.&28.	The development of algorithms and complex applications. Application of different data structures in solving problem tasks.	Presentation, individual work	Use different data types and program structures in developing more complex programs.	O5
29.&30.	Midterm exam 3	Individual work	Outcome check O4, O5	
EXERCISES/ SEMINARS				
Hours	Topic and description	Method <ul style="list-style-type: none"> • Direct teaching (lecture, instruction, pp presentation) • Discovery learning (individual, lead, discussion) • Group learning • Case study • Field classes... 	Learning outcomes	Course outcome

1. - 3.	Introduction to development tools for writing programs in C ++ language.	Lecture, discussion, PP presentation	Explain how to write a program and use the basic instructions of a development tool.	O1
4. - 6.	Development of simple programs for entering and printing data on the screen.	Presentation, independent assignment	Apply instructions for entering and printing data.	O1
7. - 9.	Variables, constants, data structures	Presentation, independent assignment	Apply different data structures to solve different types of tasks.	O2
10. - 12.	Operators	Presentation, independent assignment	Explain basic operators and create complex relations between operators.	O2
13. - 15.	Branching instructions	Presentation, independent assignment	Create simple programs using branching instructions.	O2
16. - 18.	Task - knowledge test	Individual work	Create a simple program using branch instructions together with different operators.	O2
19. - 21.	Loop instruction	Guided and individual work	Develop of simple programs applying the loop instructions.	O2
22. - 24.	Character data types and string	Guided and individual work	Apply basic functions to work with character data types and string.	O3
25. - 27.	Arrays	Guided and individual work	Apply basic functions to work with arrays.	O3
28. - 30.	Task - knowledge test	Individual work	Create a simple program.	O3
31. - 33.	Function	Guided and individual work	Develop of simple programs using a functions.	O4
34. - 36.	Vector	Guided and individual work	Develop of simple programs applying a vector data structure.	O4
37. - 39.	Task - knowledge test	Individual work	Create a simple program.	O4
40. - 42.	Files	Individual work	Develop of simple programs applying a file.	O5
43. - 45.	Task - knowledge test	Individual work	Develop of complex application.	O5

