



# POLYTECHNIC OF MEĐIMURJE IN ČAKOVEC

## COURSE SYLLABUS

ACADEMIC YEAR: 2020/2021

### 1. GENERAL COURSE INFORMATION

<b>1.1 Course name</b>	Object Oriented Programming 1			
<b>1.2 Study program/s</b>	Undergraduate professional study of <i>Computer Science</i>			
<b>1.3 Course status (O,E)</b>	Mandatory	<b>1.6 Mode of instruction (number of hours)</b>	<b>Lectures</b>	30
<b>1.4 Course code</b>	5124		<b>Exercises</b>	30
<b>1.5 Course abbreviation</b>	OOP1		<b>Seminars</b>	
<b>1.6 Semester</b>	III		<b>E-learning</b>	
<b>1.7 ECTS</b>	6	<b>1.7 Place and time of instruction</b>	The premises of the Polytechnic of Međimurje in Čakovec, according to the schedule published on the website	

### 2. TEACHING STAFF

<b>2.1 Course leader/s-title</b>	Krešimir Kolac, lecturer	<b>contact</b>	kkolac@mev.hr
		<b>contact</b>	
<b>2.2 Assistant/s- title</b>		<b>contact</b>	
		<b>contact</b>	
<b>2.3 Instruction held by- title</b>		<b>contact</b>	

### 3. COURSE DESCRIPTION

<b>3.1 Course goals</b>	Creating the skills and knowledge needed to effectively use modern object-oriented programming languages.									
<b>3.2 Prerequisites</b>	Required input competencies are the use of the procedural programming paradigm, knowledge and use of data types and control structures, knowledge and use of one-dimensional and multidimensional data fields as defined by the learning outcomes of the course Programming. The condition for taking the course is the passed subject Programming.									
<b>3.3 Course outcomes</b>	After successfully completing the course, students will be able to: I1 - Use simple (primitive, value) and complex (class-based) data types. I2 - Apply control structures of programming language. I3 - Apply classes and their components available in program libraries. I4 - Apply closure, inheritance and multiplicity. I5 - Use data storage structures. I6 - Use exception handling in program code. I7 - Use graphical user interface elements.									
<b>3.4 Course content</b>	In the course, students learn the basic concepts of object-oriented programming: classroom closure, outward interface, inheritance, and multifacetedness. The difference between procedural and object-oriented programming paradigms. Development of console and GUI desktop applications. Exception processing, control structures, data storage collections.									
<b>3.5 Types of coursework</b>	x	Lectures	x	Exercises		Blended e-learning	x	Individual activities		Laboratory

	Seminars and workshops	x	Distant learning		Field classes		Multimedia and network		Mentorship																																													
	Other																																																					
<b>3.6 Language of instruction</b>	Croatian																																																					
<b>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)</b>	2	Class attendance		Seminars		Essay																																																
		Class activity		Project		Report/paper																																																
	2	Midterm exams	2	Practical task		Continuous knowledge check																																																
		Written exam		Experimental work																																																		
		Oral exam		Research																																																		
<b>3.8 Assessment and evaluation of students' work during classes and at the final exam</b>	<p>The course has 7 defined learning outcomes.  A maximum of 200 points can be earned per course.  Learning outcomes are scored and checked through the following methods</p> <p>First intermediate exam (M1): up to 40 points, of which 20 are achieved by a theoretical test and 20 by practical work.  Second intermediate exam (M2): up to 60 points, of which 30 are achieved by a theoretical test and 30 by practical work.  Third intermediate exam (M3): up to 100 points, of which 50 are achieved by a theoretical test and 50 by practical work.</p> <p>The initial criterion for calculating the grade is expressed in the following list:</p> <ul style="list-style-type: none"> <li>● 100-125 Sufficient (2)</li> <li>● 126-150 Good (3)</li> <li>● 151-175 Very good (4)</li> <li>● 176-200 Excellent (5)</li> </ul> <p>The final criterion for calculating the grade will be created based on the Normal distribution of the total points achieved by all students in the intermediate exams M1, M2 and M3. If the newly created criterion based on the Normal distribution is less favorable for students, they will apply the Initial criterion.</p>																																																					
<b>3.9 Assessment criteria – analysis per learning outcomes</b>	<table border="1"> <thead> <tr> <th></th> <th>M1</th> <th>M2</th> <th>M3</th> <th>IN TOTAL</th> </tr> </thead> <tbody> <tr> <td>Outcome 1</td> <td>10</td> <td>10</td> <td>10</td> <td>30</td> </tr> <tr> <td>Outcome 2</td> <td>10</td> <td>10</td> <td>10</td> <td>30</td> </tr> <tr> <td>Outcome 3</td> <td>10</td> <td>10</td> <td>20</td> <td>40</td> </tr> <tr> <td>Outcome 4</td> <td></td> <td>20</td> <td>20</td> <td>40</td> </tr> <tr> <td>Outcome 5</td> <td>10</td> <td>10</td> <td>10</td> <td>30</td> </tr> <tr> <td>Outcome 6</td> <td></td> <td></td> <td>20</td> <td>20</td> </tr> <tr> <td>Outcome 7</td> <td></td> <td></td> <td>10</td> <td>10</td> </tr> <tr> <td><b>In total</b></td> <td><b>40</b></td> <td><b>60</b></td> <td><b>100</b></td> <td><b>200</b></td> </tr> </tbody> </table>										M1	M2	M3	IN TOTAL	Outcome 1	10	10	10	30	Outcome 2	10	10	10	30	Outcome 3	10	10	20	40	Outcome 4		20	20	40	Outcome 5	10	10	10	30	Outcome 6			20	20	Outcome 7			10	10	<b>In total</b>	<b>40</b>	<b>60</b>	<b>100</b>	<b>200</b>
	M1	M2	M3	IN TOTAL																																																		
Outcome 1	10	10	10	30																																																		
Outcome 2	10	10	10	30																																																		
Outcome 3	10	10	20	40																																																		
Outcome 4		20	20	40																																																		
Outcome 5	10	10	10	30																																																		
Outcome 6			20	20																																																		
Outcome 7			10	10																																																		
<b>In total</b>	<b>40</b>	<b>60</b>	<b>100</b>	<b>200</b>																																																		
<b>3.10 Specific features related with taking the course</b>	As a rule, the first midterm exam is written after the first 4 weeks of classes and covers the learning outcomes covered in the first 4 weeks. The second midterm exam is written after the other 9 weeks of classes and covers the learning outcomes covered by the ninth week of classes. The third midterm																																																					

	<p>exam is written after the second 14 weeks of teaching and covers the learning outcomes processed up to the 14th week of teaching.</p> <p>The type of questions is defined by the teacher, but all questions and tasks cover the course material or learning outcomes.</p> <p>By additional work and commitment through the preparation of homework, the student can achieve an additional amount of points by which his total amount of points does not exceed 200.</p> <p>Students who do not pass the colloquia are required to take the written and oral part of the exam. The condition for taking the oral part of the exam is passing the written part of the exam.</p>								
<b>3.11 Students obligations</b>	<p>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.</p> <p>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.</p>								
<b>3.12 Written assignments</b>									
<b>3.13 Required reading</b>	<table border="1"> <tr> <td>1.</td> <td>Julijan Štribar, Boris Motnik: Demystified C ++</td> </tr> <tr> <td>2.</td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>	1.	Julijan Štribar, Boris Motnik: Demystified C ++	2.					
1.	Julijan Štribar, Boris Motnik: Demystified C ++								
2.									
<b>3.14 Additional reading</b>	<table border="1"> <tr> <td>1.</td> <td>Internal material from lectures</td> </tr> <tr> <td>2.</td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>	1.	Internal material from lectures	2.					
1.	Internal material from lectures								
2.									
<b>4 ADDITIONAL COURSE INFORMATION</b>									
<b>4.1 Quality control</b>	<p>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.</p>								
<b>4.2 Contact the teacher</b>	<p>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.</p>								
<b>4.3 Information about the course</b>	<p>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.</p>								

<b>4.4 Course contribution to the study program</b>	<p>Apply the acquired learning skills, basic knowledge of the profession and problem solving necessary for continuing studies at a higher level.</p> <p>Apply communication and professional ethics.</p> <p>Develop programming code in multiple programming languages using modern methods and tools.</p> <p>Choose ways of structuring data in program code, as well as techniques for writing complex program forms and use standard algorithms.</p> <p>Develop applications using an object-oriented paradigm in solving program tasks.</p>
-----------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**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 of the lecture	Method of work	Lecture learning outcomes	Course learning outcome
<b>1 and 2</b>	<p>Introductory lecture, teaching methods, evaluation of student work, expected learning outcomes, literature, use of LMS system Moodle / Loomen in teaching. (1 hour)</p> <p>Introduction to Object Oriented Programming. (1 hour)</p>	<ul style="list-style-type: none"> <li>• direct teaching (presentation, instruction, pp presentation)</li> <li>• Discovery learning (independent, guided, discussion, debate)</li> <li>• Group / collaborative learning</li> <li>• case study</li> <li>• field teaching...</li> </ul>	<p>oral discussion, presentation, PP presentation</p>	<p>Explain the key differences of object-oriented programming versus procedural paradigm.</p> <p>11</p>
<b>3rd and 4th</b>	<p>Data storage mechanisms, control structures specific to the programming language used.</p>	<p>oral discussion, presentation, PP presentation</p>	<p>Use data storage structures (array sorting, other operations with storage structures)</p>	<p>15</p>
<b>5th and 6th</b>	<p>Elements of object model, abstraction, closure, modularity, hierarchy. Working with graphical user interface elements, key, input field.</p>	<p>oral discussion, presentation, PP presentation</p>	<p>Create an application user interface with basic controls</p>	<p>12, 13</p>
<b>7th and 8th</b>	<p>Concepts class and object, examples of class construction using abstraction and encapsulation.</p>	<p>oral discussion, presentation, PP presentation</p>	<p>Create a copy of the class, explain the difference between a class and an object</p>	<p>13</p>

<b>9th and 10th</b>	Verification of acquired knowledge written test that includes theoretical and programmatic knowledge examples	On its own	Outcome check I1, I2, I3, I5	I1, I2, I3, I5
<b>11th and 12th</b>	Class building, data and function members (methods), messaging between facilities.	oral discussion, presentation, PP presentation	Implement a class with data and function members.	I4
<b>13th and 14th</b>	Simple and complex data types, operations on complex data types, role of identifiers, equality of complex types based on classes, class constructor, startup constructor.	oral discussion, presentation, PP presentation	Implement class constructors, methods for type equality comparison.	I1, I4
<b>15th and 16th</b>	Relationships between classes, composition, aggregation, inheritance. Introducing relationships in program code.	oral discussion, presentation, PP presentation	Explain the basic relationships between types.	I3, I4
<b>17 and 18</b>	Program examples, problem solving	oral discussion, presentation, PP presentation	Use classes when creating an application.	I1-I5
<b>19th and 20th</b>	Verification of acquired knowledge written test that includes theoretical and programmatic knowledge examples	On its own	Outcome check I1-I5	I1-I5
<b>21st and 22nd</b>	Versatility, concept and implementation, mechanism of static and dynamic connection	oral discussion, presentation, PP presentation	Distinguish between static and dynamic linking methods	I5
<b>23rd and 24th</b>	Abstraction, closure, inheritance, multiplicity on the example of a simple application for two-dimensional vector drawing. Creating an application with a graphical user interface	oral discussion, presentation, PP presentation	Create a simple GUI application for two-dimensional drawing.	I4-I7
<b>25 - 28</b>	Principles of distribution of responsibilities among classes	oral discussion, presentation, PP presentation	Use exception handling in program code.	I3-I7
<b>29. and 30.</b>	Verification of acquired knowledge written test that includes theoretical and programmatic knowledge examples, determining the right to take the exam based on attendance records	On its own	Outcome check I1-I7	I1-I7
<b>EXERCISES</b>				
<b>Hours</b>	<b>Topic and description of the lecture</b>	<b>Method of work</b>	<b>Learning outcomes of exercises</b>	<b>Course learning outcome</b>
<b>1. - 2.</b>	Development tool, simple (primitive, value) data types, control structures, development of a program for solving two linear equations with two unknowns	presentation, independently	Create a new application in the development tool. Basic understanding of tools.	I1

<b>3. - 4.</b>	Creating a program using some of the storage structures, filling the structure with data of a simple type, performing operations using structure methods, retrieving data, copying data from one structure to another, deleting elements, sorting.	presentation, independently	Use data storage structures (array sorting, other operations with storage structures)	15
<b>5. - 6.</b>	Development of an application for calculating the sum, difference, product and quotient for two or more data.	presentation, independently	Create an application user interface with basic controls	12, 13
<b>7. - 8.</b>	Creating a simple application that uses elements of the graphical user interface, calculator.	presentation, independently	Create a complex user interface.	13
<b>9 - 10</b>	Checking the acquired skills of creating a software solution, solving the task by creating a program on a computer	on its own	Outcome check 11, 12, 13, 15	11, 12, 13, 15
<b>11. - 14.</b>	Development of a software solution based on the classes Person, Address Book	presentation, independently	Implement a class with data and function members.	11-14
<b>15. - 16.</b>	Inheritance, creating program code that uses the classes Person, Student, Professor, Records, Grade	presentation, independently	Explain the basic relationships between types.	13, 14
<b>17 - 18</b>	Inheritance, creating program code that uses the classes GraphicObject, Line, Quadrangle, Drawing Panel	presentation, independently	Explain inheritance as well	11-15
<b>19. - 20.</b>	Checking the acquired skills of creating a software solution, solving the task by creating a program on a computer	on its own	Outcome check 11-15	11-15
<b>21 - 22</b>	Implemented a programming example that uses classes linked by inheritance, to introduce a dynamic linking mechanism. Example based on the classes Person, Student, Professor.	presentation, independently	Distinguish between static and dynamic linking methods	15
<b>23. - 24.</b>	Implement a programming example that properly uses abstraction, closure, inheritance, multidisciplinary, and dynamic linking. Example based on the classes GraphicShape, Line, Quadrilateral.	presentation, independently	Create a simple GUI application for two-dimensional drawing.	14-17
<b>25 - 28</b>	Implementing a sales data management application	presentation, independently	Use exception handling in program code.	13-17
<b>29. - 30.</b>	Checking the acquired skills of creating a software solution, solving the task by creating a program on a computer	on its own	Outcome check 11-17	11-17