MODULE DESCRIPTION

Module code	full-time studies:	**Z-ZIP1-E-621**
	part-time studies:	**Z-ZIPN1-E-621**
Module name	**RAD Object Oriented Programming**	
Module name in Polish	**Programowanie obiektowe w RAD**	
Valid from academic year	**2019/2020**	

MODULE PLACEMENT IN THE SYLLABUS

Field of study	**MANAGEMENT AND PRODUCTION ENGINEERING**
Level of education	**1st degree**
Studies profile	**General**
Form and method of conducting classes	**Full-time and Part-time**
Specialisation	**Computer Science for Management and Modelling**
Unit conducting the module	**Department of Computer Science Technologies**
Module co-ordinator	**Sławomir Koczubiej, PhD**
Approved by:	**Dariusz Bojczuk, PhD, DSc**

MODULE OVERVIEW

Type of subject / group of subjects	**Specialist subject**
Module status	**Non-compulsory**
Language of conducting classes	**English**
Module placement in the syllabus - semester	**Semester VI**
Initial requirements	**Information Technologies,** **Fundamentals of Computer Science** **Algorithms and Data Structures**
Examination (YES/NO)	**YES**
Number of ECTS credit points	**3**

Method of conducting classes		**Lecture**	**Classes**	**Laboratory**	**Project**	**Other**
Per semester	full-time studies:	**15**		**30**		
	part-time studies:	**9**		**18**		

TEACHING RESULTS AND THE METHODS OF ASSESSING TEACHING RESULTS

Category	Symbol	Learning outcomes	Assignations to the directional learning outcomes
Knowledge	W01	A student has knowledge of the syntax and semantics of control instructions in an object-oriented language. The student has knowledge of data types and the selection of these types to solve a specific programming task. Understands the concepts of inheritance, aggregation and polymorphism of objects.	ZIP1_W05
	W02	A student has an extended knowledge of programming in the field of using files. He knows the rules of handling files in accordance with the architecture of the operating system.	ZIP1_W04 ZIP1_W05
	W03	A student has knowledge of the principles of using graphic components to build the application GUI. Has knowledge of the principles of using the RAD type tool for designing multi-window applications.	ZIP1_W05
Skills	U01	A student is able to write in a programming language an object-oriented model describing a simple engineering problem.	ZIP1_U07
	U02	A student is able to use the programming environment to design and build a computer program, compile, consolidate and test the program. A student is able to design and build a computer program, using ready-made objects of the programming system with the use of files and graphic illustrations.	ZIP1_U07
	U03	A student is able to analyze a simple source code, diagnose errors and introduce modifications to the content of an existing program.	ZIP1_U07
Social competences	K01	A student understands the need for constant replenishment of knowledge. Has competences in the use of Internet resources for self-education.	ZIP1_K01
	K02	A student is ready to work individually and in a team (by accepting various roles in it).	ZIP1_K04

TEACHING CONTENTS

Method of conducting classes	Teaching contents
Lecture	Introduction to an object-oriented language. Variables and types. Control statements, arrays and lists. Classes, objects, methods. Exceptions and their handling. Objects and memory management. Object creation and destruction. Operations on texts, special characters, string processing. Console applications with parameters. Overloaded operators. Streams and file support. Graphic interface, structure, tasks.
Laboratory	Conditional and iterative processing. Building an object-oriented application. Defined classes. Class members, objects. Inheritance, polymorphism and pointer arrays. Object creation and destruction. Constructor, destructor and memory management. Applications processing strings. Building a console application with parametric calling. Development of an application with a graphical user interface. Event driven applications. Cooperation with files. Multi-window applications.

METODS OF ASSESSING TEACHING RESULTS

Symbol	**Methods of checking the learning outcomes** *(select X)*					
	Oral exam	**Written exam**	**Test**	**Project**	**Statement**	**Other**
W01		X	X			
W02		X	X			
W03		X	X			
U01		X	X			X
U02		X	X			X
U03		X	X			X
K01						X
K02						X

FORM AND CONDITIONS OF PASSING

Form of classes	**Form of credit**	**Passing conditions**
Lecture	Exam	Obtaining at least 50% of the exam points, lecture comments.
Laboratory	Credit with grade	Obtaining at least 50% of tests, active participation in laboratory classes.

STUDENT WORKLOAD

Balance of ECTS points												
No.	**Type of student's activity**	**Student's workload**										**Unit**
		full-time					**part-time**					
		Lc	C	Lb	P	O	Lc	C	Lb	P	O	
1.	Participation in the activities	15		30			9		18			h
2.	Other (consultation, exam)	4		2			4		2			h
3.	**Number of hours of a student's assisted work**	**51**					**33**					h
4.	**Number of ECTS credit points which are allocated for assisted work**	**2,0**					**1,3**					ECTS
5.	**Number of hours of a student's unassisted work**	**49**					**67**					h
6.	**Number of ECTS credit points which a student receives for unassisted work**	**2,0**					**2,7**					ECTS
7.	**Work input connected with practical classes**	**67**					**67**					h
8.	**Number of ECTS credit points which a student receives for practical classes**	**2,7**					**2,7**					ECTS
9.	**Total number of hours of a student's work**	**100**					**100**					h
10.	**Punkty ECTS za moduł** *1 ECTS=25 hours*	**4**										ECTS

LITERATURE

1. Prata S. (2013), *C Primer Plus*, Addison-Wesley Professional.
2. Prata S. (2013), *C++ Primer Plus*, Addison-Wesley Professional.
3. Summerfield M. (2013), *Advanced Qt Programming*, Financial Times Prentice Hall, 2013.
4. Troelsen A., Japikse P. (2022), *Pro C# 10 with .NET. 6: Foundational Principles and Practices in Programming*, Apress.