MODULE DESCRIPTION

Module code	**Z-ZIP-611z**
Module name	**Języki programowania – C++**
Module name in English	**Programming Languages – C++**
Valid from academic year	**2016/2017**

A. MODULE PLACEMENT IN THE SYLLABUS

Field of study	**Management and Production Engineering**
Level of education	**1st degree** *(1st degree / 2nd degree)*
Studies profile	**General** *(general / practical)*
Form and method of conducting classes	**Full-time** *(full-time / part-time)*
Specialisation	**All**
Unit conducting the module	**Department of Applied Computer Science and Applied Mathematics**
Module co-ordinator	**Marzena Nowakowska, PhD hab.**
Approved by:	

B. MODULE OVERVIEW

Type of subject/group of subjects	**Major** *(basic / major / specialist subject / conjoint / other HES)*
Module status	**Non-compulsory** *(compulsory / non-compulsory)*
Language of conducting classes	**English**
Module placement in the syllabus - semester	**5th semester**
Subject realisation in the academic year	**Winter semester** *(winter / summer)*
Initial requirements	**Fundamentals Computer Science** *(module codes / module names)*
Examination	**No** *(yes / no)*
Number of ECTS credit points	**3**

Method of conducting classes	Lecture	Classes	Laboratory	Project	Other
Per semester	**15**		**24**		

C. TEACHING RESULTS AND THE METHODS OF ASSESSING TEACHING RESULTS

Module target	The aim of the module is to gain the ability to build a computer program as regards the following: C++ language statements and basic data structures; programming in the graphical environment of the Borland C++ Builder (BCB) system using the functions of handling events; and the principles of using components offered in standard BCB palettes.

Effect symbol	Teaching results	Teaching methods *(l/c/lab/p/other)*	Reference to subject effects	Reference to effects of a field of study
W_01	A student knows and understands the principles of functioning of computer programs and managing them in Windows environment.	l/lab	K_W05	T1A_W03 S1A_W06
W_02	A student has basic knowledge as regards data types and selecting those types to solve a specific programming task.	l/lab	K_W05	T1A_W03 S1A_W06
W_03	A student understands a modular structure of a computer program and the need of utilizing built-in functions of the BCB environment and his/her own programmer's functions.	l/lab	K_W05	T1A_W03 S1A_W06
U_01	A student is able to design and build a window application using ready objects of the BCB programming system as well as his/her own programming solutions.	lab	K_U07	TA1_U01 TA1_U07 TA1_U08
U_02	A student is able to make the analysis of a simple source code as well as to introduce modifications in the code of an existing program.	l/lab	K_U07	TA1_U01 TA1_U07 TA1_U08
U_03	A student has the ability of defining his/her own programmer's functions as well as utilising them in a computer application created by him/her.	lab	K_U07	TA1_U01 TA1_U07 TA1_U08
U_04	A student has the skills of elaborating algorithms to solve various programming tasks according to the principles of universal logic.	lab	K_U07	TA1_U01 TA1_U07 TA1_U08
K_01	A student understands the necessity of continuous usage and enrichment of his/her knowledge as regards algorithmic operations.	l/lab	K_K01	T1A_K01
K_02	A student is able to work individually and in a group (by accepting diverse roles).	l	K_K04	T1A_K03 T1A_K04

Teaching contents:

1. Teaching contents as regards lectures

Lecture number	Teaching contents	Reference to teaching results for a module
1-2	The IDE of the Borland C++ Builder system. Program structure in the C++ programming language. Application structure in BCB. Creating a source code. Component palette, communication of an application with a user. Basic language statements.	W_01 W_02 U_01 U_04 K_01
3	Simple data types. Selected operators, their precedence and operational sequence. The algorithms of iterative processing. Arrays and loop statements.	W_02 U_04 K_02
4-5	Pointer types. Pointer and dereference operators. Address arithmetic. Working with arrays. Control in the program using the loop statements.	W_03 U_02 U_03 K_02

6	Defining functions and passing parameters to them. The elements of objective programming – using some example BCB components. The properties and methods of a class as tools used when utilising objects in a computer program. The *AnsiString* class: properties and methods.	W_03 U_03 U_04 K_01
7	Text processing; *TMemo* and *TStrings* classes. Organising the access to a text file using dialogue components.	W_03 U_02
8	The co-operation of an application with a text file. The role of the *TStrings* class in data transmission between the application and the text file.	W_03 U_02 U_03 K_02

2. Teaching contents as regards classes

Class number	Teaching contents	Reference to teaching results for a module

3. Teaching contents as regards laboratory classes

Laboratory class number	Teaching contents	Reference to teaching results for a module
1-2	The structure of a project in the Borland C++ Builder (BCB) system. Communicating an application with a user. Standard object chart in BCB. Data type conversion and arithmetic operations. Control in a computer program.	
3-5	The sequence of statement execution within a computer program. Calculation algorithms. Iterative processing. Implementing calculation algorithms in the BCB application. Arrays and loop statements: calculating statistics from numerical arrays. Test 1.	
6-7	Iterative array processing using the *TStringGrid* class to enter data and derive results. User function as a class component and as an external function. Global variables. Passing function parameters.	
8-9	String processing: the *AnsiString* class. Iterative text processing: statistics, searching and modifying texts. Test 2.	
10-12	Organising access to text files. Data transmission between disk and main memory (array structures and visual form components). Creating applications co-operating with a text file. Test 3.	

4. The characteristics of project assignments

The methods of assessing teaching results

Effect symbol	Methods of assessing teaching results *(assessment method, including skills – reference to a particular project, laboratory assignments, etc.)*
W_01	Individual tasks during classes.
W_02	Test 2 during the classes
W_03	Test 2 during the classes
U_01	Test 1 during the classes: a student's own application providing calculations. Work with the

	use of a computer.
U_02	Commenting and discussing students' solution during the classes and lectures.
U_03	Test 3 during the classes: a student's own application co-operating with a disk file. Work with the use of a computer.
U_04	Test 1 and 3 during the classes: a student's own application providing calculations and co-operating with a disk file respectively. Work with a computer.
K_01	Commenting and discussing students' solution during the classes and lectures.
K_02	Observing a student's involvement during laboratory classes and a discussion of problems reported by students.

D. STUDENT'S INPUT

ECTS credit points		
	Type of student's activity	**Student's workload**
1	Participation in lectures	**15**
2	Participation in classes	
3	Participation in laboratories	**24**
4	Participation in tutorials (2-3 times per semester)	**3**
5	Participation in project classes	
6	Project tutorials	
7	Participation in an examination	
8		
9	**Number of hours requiring a lecturer's assistance**	**42** *(sum)*
10	**Number of ECTS credit points which are allocated for assisted work** *(1 ECTS point=25-30 hours)*	**1.4**
11	Unassisted study of lecture subjects	**10**
12	Unassisted preparation for classes	
13	Unassisted preparation for tests	**14**
14	Unassisted preparation for laboratories	**24**
15	Preparing reports	
15	Preparing for a final laboratory test	
17	Preparing a project or documentation	
18	Preparing for an examination	
19		
20	**Number of hours of a student's unassisted work**	**48** *(sum)*
21	**Number of ECTS credit points which a student receives for unassisted work** *(1 ECTS point=25-30 hours)*	**1.6**
22	**Total number of hours of a student's work**	**90**
23	**ECTS points per module** *1 ECTS point=25-30 hours*	**3**
24	**Work input connected with practical classes** *Total number of hours connected with practical classes*	**60**
25	**Number of ECTS credit points which a student receives for practical classes** (1 ECTS point=25-30 hours)	**2**

E. LITERATURE

Literature list	1. Any Internet source on C++ programming language and Borland C++ Builder
Module website	