Załącznik Nr 5 do Zarz. Nr 33/11/12

Z1-PU7	WYDANIE N1	Strona 1 z 3

(*faculty stamp*)

COURSE DESCRIPTION

1. Course title: COMPUTER PROGRAMMING	**2. Course code** CP (CP2 and CP3)
3. Validity of course description: 2017/2018	
4. Level of studies: 1st cycle of higher education	
5. Mode of studies: intramural studies	
6. Field of study: Macrofaculty	RAU
7. Profile of studies:	
8. Programme:	
9. Semester: II, III	
10. Faculty teaching the course: Faculty of Automatic Control, Electronics and Computer Science	
11. Course instructor: dr inż. Piotr Fabian, dr inż. Roman Starosolski	
12. Course classification: common courses	
13. Course status: compulsory	
14. Language of instruction: English	
15. Pre-requisite qualifications: completed course: Fundamentals of Computer Programming	

16. Course objectives: During the course students acquire knowledge necessary to understand and design computer programs as well as skills of implementing programs with structural and object-oriented languages. The objective of the course is to give solid grounds, both theoretical and practical, in the field of software development. Laboratory classes allow to exploit in practice informations acquired during lectures, from basic knowledge to advanced techniques, in particular related to Object Oriented Programming and Generic Programming. C and C++ languages including C++11 and STL are used during the course.

17. Description of learning outcomes:

Nr	Learning outcomes description	Method of assessment	Teaching methods	Learning outcomes reference code
1	Has a detailed knowledge in the field of Structural Programming and basic in Object Oriented programming	Written test (sem. 2)	Lecture, Laboratory	K1A_W3, K1A_W12
2	Can perform practical programming tasks exploiting selected Structural and Object Oriented programming techniques	Computer program (sem. 2)	Laboratory	K1A_U10, K1A_U12
3	Can develop nontrivial computer programs	Computer program (sem. 2)	Laboratory	K1A_U3, K1A_U10, K1A_W12,
4	Has a detailed knowledge in the field of Object Oriented programming	Written test (sem. 3)	Lecture, Laboratory	K1A_W3, K1A_W12
5	Can find, broaden and present information on selected Object Oriented programming technique	Presentation (sem. 3)	Laboratory	K1A_U1, K1A_U2, K1A_K1, K1A_K3
6	Can perform practical programming tasks exploiting selected Object Oriented programming techniques	Computer program (sem. 3)	Laboratory	K1A_U10, K1A_U12
7	Can construct complex Object Oriented software based on project technical documentation	Computer program (sem. 3)	Laboratory	K1A_U3, K1A_U10, K1A_W12,

18. Teaching modes and hours

Lecture / BA /MA Seminar / Class / Project / Laboratory

Sem. 2 - Lecture 15 h., Laboratory 15 h.; Sem. 3 - Lecture 30 h., Laboratory 30 h

19. Syllabus description:

Lecture:

Introduction: first sample programs. Variables, fundamental and complex types, operators and expressions. Instructions and flow control. Functions, program structure, memory classes. Arrays, pointers and dynamic memory management. Structures, unions and dynamic data structures. Preprocessor, header files and libraries. Object oriented programming paradigm. Object oriented programming on the example of the C++ standard language. Overview of object oriented programming. Classes, objects, class methods. Friend methods and classes. Constructors and destructors. Life cycle of objects. Overloading of operators; user-defined type conversions. Inheritance and polymorphism. Abstract classes. Multiple inheritance and virtual base classes. Exceptions. Run-Time Type Information. Templates. Standard libraries: streams, strings, elements of the Standard Template Library and generic programming. C++11.

Laboratory:

Basic types, operators, instructions. Array and pointer types. Libraries and library functions. Working with files. Structures & unions. Not OOP-oriented C++ features. Classes, constructors and destructors. Overloading of operators. Inheritance. Multiple inheritance. Polymorphism. RTTI. Streams. Exceptions. Templates. Design and implementation of own programming task. Second half of the 3rd semester is devoted to developing of individual programming task, while earlier laboratories consist mainly of shorter tasks allowing to learn and practice selected programming techniques.

20. Examination: —

21. Primary sources:

B. Stroustrup, The C++ Programming Language. Addison-Wesley, Reading, MA.
ISO/IEC International Standard for Programming Language C++, free working draft, 2011 (http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2011/n3242.pdf)
Nicolai M. Josuttis: C++ Standard Library: A tutorial and Reference, 2nd, Addison Wesley Longman 2012

22. Secondary sources:

The Computer Programming 2 course (http://platforma.polsl.pl/rau2/course/view.php?id=89)
The Computer Programming 3 course (http://platforma.polsl.pl/rau2/course/view.php?id=88)
The C++ Resources Network (http://www.cplusplus.com/)
The Bjarne Stroustrup's homepage with C++11 FAQ (http://www.stroustrup.com/index.html)
The C++11 wikipedia page (http://en.wikipedia.org/wiki/C%2B%2B11)
Grębosz J.: Symfonia C++. RM, W-wa, wyd. 4 (in Polish only)
Grębosz J.: Pasja C++. RM, W-wa, wyd. 2 (in Polish only)

23. Total workload required to achieve learning outcomes

Lp.	Teaching mode :	Contact hours / Student workload hours
1	Lecture	45 / 30
2	Classes	- / -
3	Laboratory	45 / 60
4	Project	/
5	BA/ MA Seminar	/
6	Other	- / 30
	Total number of hours	90 / 120

24. Total hours: 210

25. Number of ECTS credits: 7 (3 - sem. 2, 4 - sem. 3)

26. Number of ECTS credits allocated for contact hours: 3

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 2
26. Comments: —

Approved:

…………………………….
(date, Instructor's signature)

……………………………………………………
(*date , the Director of the Faculty Unit signature)*