

(pieczęć wydziału)

**COURSE DESCRIPTION**

Z1-PU7

WYDANIE N1

Strona 1 z 4

<b>1. Course title:</b> COMPUTER PROGRAMMING		<b>2. Course code:</b> CP		
<b>3. Validity of course description from:</b> 2018/2019				
<b>4. Level of studies:</b> 1 <sup>st</sup> cycle of higher education				
<b>5. Mode of studies:</b> intramural studies				
<b>6. Field of study:</b> INFORMATYKA (RAU)				
<b>7. Profile of studies:</b> general academic				
<b>8. Specialization:</b> -				
<b>9. Semester:</b> II, III, and IV				
<b>10. Faculty teaching the course:</b> Wydział Automatyki, Elektroniki i Informatyki				
<b>11. Course instructors:</b> dr hab. inż. Sebastian Deorowicz, prof. Pol. Śl.; dr inż. Jacek Lach; dr inż. Dariusz Myszor; dr inż. Roman Starosolski				
<b>12. Course classification:</b> common courses				
<b>13. Course status:</b> compulsory				
<b>14. Language of instruction:</b> English				
<b>15. Pre-requisite qualifications:</b> completed course: Fundamentals of Computer Programming				
<b>16. Course objectives:</b> The objective of the course is to provide knowledge necessary to understand the design and implementation of programs as well as the skills to implement the software on the example of the C, C++ and C# languages. The aim of the course is to create strong theoretical and practical foundations in the field programming paradigms: structural, object-oriented, generic as well as metaprogramming. The laboratory and the project allow to apply in practice the knowledge acquired during lectures, including knowledge necessary to create software, in particular concerning basic and advanced object-oriented techniques on the example of the C++ language and elements of C# language (object-oriented analysis, design, and programming).				
<b>17. Efekty kształcenia:<sup>1</sup></b>				
Nr	Learning outcomes description	Method of assessment	Teaching methods	Learning outcomes reference code
1	Ability to perform practical programming tasks exploiting selected programming techniques	Computer program (sem. 2)	Laboratory	K1A_W09, K1A_W11, K1A_W12, K1A_W22,

<sup>1</sup> należy wskazać ok. 5 – 8 efektów kształcenia

2	Ability to create software using structural programming	Computer program (sem. 2)	Laboratory	K1A_W09, K1A_W10, K1A_W11, K1A_W12, K1A_W14, K1A_W15, K1A_W22, K1A_U03, K1A_U07, K1A_U20, K1A_U21, K1A_U23, K1A_U26, K1A_U29
3	Ability to apply basic and intermediate structural programming techniques	Written test (sem. 2)	Lecture, laboratory	K1A_W09, K1A_W11, K1A_W12,
4	Ability to perform practical programming tasks exploiting selected programming techniques	Computer program (sem. 3)	Laboratory	K1A_W09, K1A_W11, K1A_W12, K1A_W22,
5	Ability to create software using the available technical documentation	Computer program (sem. 3)	Project	K1A_W09, K1A_W10, K1A_W11, K1A_W12, K1A_W14, K1A_W15, K1A_W22, K1A_U03, K1A_U07, K1A_U20, K1A_U21, K1A_U26, K1A_U29,
6	Ability to apply basic and intermediate object-oriented programming techniques	Written test (sem. 3)	Lecture, laboratory	K1A_W09, K1A_W11, K1A_W12,
7	The ability to find, explore, and present information on a selected subject in the field of object-oriented programming	Presentation (sem. 4)	Laboratory	K1A_W09, K1A_W11, K1A_W13, K1A_U01, K1A_U02, K1A_K01, K1A_K02, K1A_K03, K1A_U04
8	Ability to perform practical programming tasks exploiting selected advanced object-oriented programming techniques	Computer program (sem. 4)	Laboratory	K1A_W09, K1A_W10, K1A_W11, K1A_W12, K1A_W13, K1A_W22,
9	Ability to create complex software using the available technical documentation	Computer program (sem. 4)	Project	K1A_W09, K1A_W10, K1A_W11, K1A_W12, K1A_W13, K1A_W14, K1A_W15, K1A_W22, K1A_U03, K1A_U04, K1A_U07, K1A_U20, K1A_U21, K1A_U26, K1A_U29, K1A_U30
10	Has a detailed knowledge in the field of structural and object-oriented programming	Exam (sem. 4)	Lecture, laboratory, project	K1A_W09, K1A_W11, K1A_W12, K1A_W13, K1A_K01

### 18. Teaching modes and hours

**Lecture 60    Laboratory 45    Project 15**

### 19. Syllabus description:

#### Lecture:

The C language preprocessor. Application development in C language: multithreaded applications, regular expressions, state machine. Good practice of software development in C. Tools supporting the work of

programmer in C. Introduction to C++, non-object-oriented elements of C++. Object-oriented techniques on the example of the C++ language. Paradigm of object-oriented programming, abstract data types. Classes, objects, life cycle of objects. Operator overloading. Inheritance. Polymorphism. Run time type information. Abstract classes. Multiple inheritance. The mechanism of exceptions. Templates. The C++ language library, the C++ standard library, the input/output library. The generic programming; The STL library, containers, iterators and algorithms. Smart pointers. Metaprogramming. Parallelism. Selected and advanced elements of the standard library. Introduction to the C# language, basic types, dynamic memory management, description of technologies, reflection mechanism, operators, inheritance, generic types.

**Laboratories and projects:**

The data types, operators, instructions. Array and pointer types. Libraries and library functions. Working with files. Structures and unions. Implementation of selected tasks (eg., state machine, cyclic buffer). Non object-oriented C++ elements. Classes, constructors, and destructors. Operator overloading. Inheritance. Polymorphism. The mechanism of exceptions. RTTI. Templates. STL containers, iterators, and algorithms. Smart pointers. Threads (basics). Regular expressions. Implementation of complex programming tasks in C, C++ and C#.

**20. Egzam: yes**

**21. Primary sources:**

- B. W. Kernighan, D.M.Ritche, The C Programming Language (ANSI C), Patience Hall, NY
- ISO/IEC JTC1/SC22/WG14 International Standard—Programming Languages C (draft C11: <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1539.pdf>)
- B. Stroustrup, The C++ Programming Language. Addison-Wesley, Reading, MA
- ISO/IEC JTC1/SC22/WG21 International Standard—Programming Languages C++ (draft C++14: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3690.pdf>)
- T. Dimes, C# Programming for Beginners: An Introduction and Step-by-Step Guide to Programming in C#

**22. Secondary sources:**

- K. N. King, C Programming: A Modern Approach, 2nd Edition, W. W. Norton & Company.
- R. Reese, Understanding and Using C Pointers: Core Techniques for Memory Management, O'Reilly Media.
- Nicolai M. Josuttis: The C++ Standard Library: A Tutorial and Reference (2nd Edition), Addison-Wesley Professional
- B. Stroustrup, The C++ Programming Language
- **Grębosz J.: Symfonia C++. RM, W-wa, wyd. 4 (in Polish only)**
- **Grębosz J.: Pasja C++. RM, W-wa, wyd. 2 (in Polish only)**
- Troelsen, Pro C# 5.0 and the .NET 4.5 Framework

**23. Total workload required to achieve learning outcomes**

Lp.	Teaching mode :	Contact hours / Student workload hours
1	Lecture	60 / 30
2	Classes	- / -
3	Laboratory	45 / 45
4	Project	15 / 45
5	Seminar	- / -
6	Other	- / 30
	Total	120 / 150

<b>24. Total hours:</b> 270
<b>25. Number of ECTS credits:</b> <sup>2</sup> 3 (sem.II), 4 (sem.III), 2, (sem.IV)
<b>26. Number of ECTS credits allocated for contact hours:</b> 4
<b>27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects):</b> 2
<b>26. Comments:</b> -

Approved:

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

.....  
 (date, the Director of the Faculty Unit signature)

---

<sup>2</sup> 1 punkt ECTS – 30 godzin.