1. Course title: FUNDAMENTALS OF COMPUTER PROGRAMMING
2. Course code: FCP
4. Level of studies: 1st cycle of higher education
5. Mode of studies: intramural studies
6. Field of studies: INFORMATICS
7. Profile of studies: general academic
8. Specialty: -
9. Semester: 1
10. Faculty teaching the course: Faculty of Automatic Control, Electronics, and Computer Science
11. Course instructor: dr hab. inż. Krzysztof Simiński
12. Course classification: common courses
13. Course status: obligatory
14. Language of instruction: English
15. Prerequisite qualification: –
16. Course objectives:
The objective to the course is introduction into design, implementation, and testing of software in high-level programming language (C++). This is a propedeutic course for object programming, software engineering, and algorithms and data structures.
17. Description of learning outcomes:

<table>
<thead>
<tr>
<th>No.</th>
<th>learning outcomes description</th>
<th>method of assessment</th>
<th>teaching methods</th>
<th>kearing outcomes reference codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student knows programming paradigms, software design methods.</td>
<td>written exam</td>
<td>lecture</td>
<td>K1A_W09</td>
</tr>
<tr>
<td>2</td>
<td>Student knows syntax and semantics of high-level programming language</td>
<td>project, laboratory</td>
<td>lecture, laboratory</td>
<td>K1A_W11</td>
</tr>
<tr>
<td>3</td>
<td>Student can write down an algorithm in a high-level programming language.</td>
<td>project, written exam</td>
<td>lecture, laboratory</td>
<td>K1A_W12, K1A_W15, K1A_W17</td>
</tr>
<tr>
<td>4</td>
<td>Student can create software and can use technical manuals.</td>
<td>project</td>
<td>laboratory</td>
<td>K1A_W15, K1A_W17, K1A_U03, K1A_U07</td>
</tr>
<tr>
<td>5</td>
<td>Student can test software.</td>
<td>project</td>
<td>laboratory</td>
<td>K1A_U15, K1A_U19</td>
</tr>
</tbody>
</table>
18. Teaching modes and hours
   lecture: 30   laboratory: 30
19. Syllabus description:
Lecture:
1. introductory issues: object, operations, programs, processes, programming paradigms
2. conditional statements, loops
3. arrays
4. functions, recursion
5. files and streams
6. structures
7. pointers, allocation and deallocation of memory
8. singly and doubly linked lists
9. binary search trees

Laboratory:
1. introductory issues: data types, variables, instructions, conditions, simple program
2. loops, array, minimum (maximum) search, sorting
3. functions, parameters, return values
4. structures
5. files and streams
6. dynamic abstract structures (lists and trees)

Project
20. Written exam: yes

21. Primary sources:
   1. A. Allain: „Jumping into C++”
   2. N. Wirth: „Algorithms + Data Structures = Programs”

22. Secondary sources:
   1. B. Stroustrup: „The C++ Programming Language”, 2014

23. Total workload required to achieve learning outcomes

<table>
<thead>
<tr>
<th>No.</th>
<th>teaching mode</th>
<th>contact hours / student workload hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lecture</td>
<td>30 / 30</td>
</tr>
<tr>
<td>2</td>
<td>classes</td>
<td>/</td>
</tr>
<tr>
<td>3</td>
<td>laboratory</td>
<td>30 / 60</td>
</tr>
<tr>
<td>4</td>
<td>project</td>
<td>- / -</td>
</tr>
<tr>
<td>5</td>
<td>BA/MA seminar</td>
<td>- / -</td>
</tr>
<tr>
<td>6</td>
<td>other (exam)</td>
<td>- / 30</td>
</tr>
<tr>
<td></td>
<td>total number of hours</td>
<td>60 / 120</td>
</tr>
</tbody>
</table>

24. Total hours: 180

25. Number of ECTS credits: 6

26. Number of ECTS credits for contact hours: 2

27. Number of ECTS credits for in-practice hours (laboratory, classes, project): 2

26. Comments: -
approved:

................................................................................................................
(date, instructor's signature) ..............................................................................
(date, signature of the Director of the Faculty Unit)