1. **Course title:** PROGRAMMING IN ASSEMBLER
2. **Course code:** PiA

3. **Validity of course description:** 2017/2018

4. **Level of studies:** BA, BSc programme / MA, MSc programme

5. **Mode of studies:** intramural studies / extramural studies

6. **Field of study:** CEIE - Interdisciplinary Studies: Automatic Control and Robotics, Electronics and Telecommunications, Computer Science (RAU)

7. **Profile of studies:** comprehensive / practical

8. **Programme:** Informatics

9. **Semester:** 1

10. **Faculty teaching the course:** Faculty of Automatic Control, Electronics and Computer Science

11. **Course instructor:** Ph.D. Eng. Krzysztof Tokarz, Ph.D. Eng. Piotr Czekalski

12. **Course classification:** common courses

13. **Course status:** compulsory / elective

14. **Language of instruction:** English

15. **Pre-requisite qualifications:** Microprocessor Systems, Theory of logic circuits, Digital circuits.

16. **Course objectives:** The goal of the topic is to teach students basic knowledge and skills of low level programming. Knowing assembler programming language will help students to make better choices of programming tools to accomplish the tasks that require time or memory optimization. It helps to get good knowledge about functioning of the processor and whole computer and conscious usage of high level programming languages.

17. **Description of learning outcomes:**

<table>
<thead>
<tr>
<th>Nr</th>
<th>Learning outcomes description</th>
<th>Method of assessment</th>
<th>Teaching methods</th>
<th>Learning outcomes reference code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student has basic knowledge of Programming in assembly language</td>
<td>Lab report</td>
<td>Lecture, laboratory</td>
<td>K2A_W29</td>
</tr>
<tr>
<td>2</td>
<td>Student knows methods, tools and techniques used to write, analyze and optimize algorithms written in assembly language.</td>
<td>Lab report</td>
<td>Lecture, laboratory</td>
<td>K2A_W10</td>
</tr>
<tr>
<td>3</td>
<td>Student has the knowledge of the time of life of modern processors.</td>
<td>Lab report</td>
<td>Lecture, laboratory</td>
<td>K2A_W25</td>
</tr>
<tr>
<td>4</td>
<td>Student can communicate in engineers community also in English</td>
<td>Lab report</td>
<td>Lecture, laboratory</td>
<td>K2A_U02</td>
</tr>
<tr>
<td>5</td>
<td>Student can optimize programs written in assembly language</td>
<td>Lab report</td>
<td>Lecture, laboratory</td>
<td>K2A_U08</td>
</tr>
<tr>
<td>6</td>
<td>Student can write assembly software based on the specification given.</td>
<td>Lab report</td>
<td>Lecture, laboratory</td>
<td>K2A_U23</td>
</tr>
</tbody>
</table>
7. Student can asset usefulness of low level programming technique for specific tasks.

Lab report

Laboratory

K1A_U18

18. Teaching modes and hours:
Lecture / BA / MA Seminar / Class / Project / Laboratory:
15 / 0 / 0 / 0 / 0 / 15

19. Syllabus description:
Lecture:

1. Introduction. Place of assembly language in modern computer science and in programming languages hierarchy. Assembler programs: MASM, MASM32 and others.
3. Procedure calling, interrupt handling, exceptions, stack.
4. Basic and advanced data types, defining of variables.
5. Instruction set of x86 family of processors. Format of the instruction.
8. Segmentation of memory and segments in assembler program. Segments definition, data structures, records, strings, repeat blocks.
9. Symbols, operators, expressions, predefined symbols.
10. Conditional assembling, macros, connections between modules.
11. Writing mixed language programs, assembler with high level languages: C, Pascal, Basic.
12. Optimization of programs, writing dll libraries.

Laboratory:

1. Simple program in MS Windows system.
2. Structure of the program with one main window. Message boxes.
3. Calling MS Windows functions from assembler program.
4. Assembler program in Visual Studio, writing assembler modules with C program.

20. Examination: no
21. Primary sources:
   a) Volume 1: Basic Architecture
   b) Volume 2: Instruction Set Reference
   c) Volume 3: System Programming Guide.

22. Secondary sources:

23. Total workload required to achieve learning outcomes:

<table>
<thead>
<tr>
<th>Lp.</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>30/55</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>/</td>
</tr>
<tr>
<td>5</td>
<td>Seminar</td>
<td>/</td>
</tr>
<tr>
<td>6</td>
<td>Others</td>
<td>5/-</td>
</tr>
<tr>
<td></td>
<td>Total number of hours</td>
<td>65/85</td>
</tr>
</tbody>
</table>

24. Total hours: 120

25. Number of ECTS credits: 5

26. Number of ECTS credits allocated for contact hours: 2

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 3

26. Comments:

Approved: