(Faculty stamp) **COURSE DESCRIPTION**

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:	

Nr	Learning outcomes description	Method of assessment	Teaching methods	Learning outcomes reference code
1	Student has basic knowledge of Programming in assembly language	Lab report	Lecture, laboratory	K2A_W29
2	Student knows methods, tools and techniques used to write, analyze and optimize algorithms written in assembly language.	Lab report	Lecture, laboratory	K2A_W10
3	Student has the knowledge of the time of life of modern processors.	Lab report	Lecture, laboratory	K2A_W25
4	Student can communicate in engineers community also in English	Lab report	Lecture, laboratory	K2A_U02
5	Student can optimize programs written in assembly language	Lab report	Lecture, laboratory	K2A_U08
6	Student can write assembly software based on the specification given.	Lab report	Lecture, laboratory	K2A_U23

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.
2. Architecture of x86 family of processors. Programming model of processors. History of Intel processors from 8086 to Core i7. Registers, flags, memory organization, logical, physical, linear addresses, addressing modes, I/O addressing.
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.
6. Math coprocessor, SIMD technology instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4, AVX. Instructions of AMD processors.
7. Assembler language characteristic. Basic elements of language: constants, symbols, statements, keywords, directives.
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:

1. Wróbel Eugeniusz „Praktyczny kurs asemblera. Wydanie II", Wydawnictwo Helion, Gliwice 2011.
2. Wróbel Eugeniusz „Programowanie w języku asemblera MASM. Laboratorium", Wydawnictwo Politechniki Śląskiej, Gliwice 2005.
3. Randal Hyde „Asembler. Sztuka programowania. Wydanie II", Wydawnictwo Helion, Gliwice 2010.
4. Wróbel Eugeniusz (pod red.) „Asembler. Ćwiczenia praktyczne", Wydawnictwo Helion, Gliwice 2002.
5. Vlad Pirogov „Asembler. Podręcznik programisty", Wydawnictwo Helion, Gliwice 2005.
6. Microsoft Macro Assembler Programmer's Guide (available in the Internet).
7. IA-32 Intel Architecture. Software Developer's Manual (available at www.intel.com):

a) Volume 1: Basic Architecture

b) Volume 2: Instruction Set Reference

c) Volume 3: System Programming Guide.

22. Secondary sources:

1. Sivarama P. Dandamudi: "Introduction to Assembly Language Programming", Springer Verlag 1998.
2. Karen Miller "An Assembly Language Introduction to Computer Architecture", Oxford University Press 1999.
3. Eugeniusz J. Wróbel: „Asembler 8086/88", Seria "Mikrokomputery", WNT Warszawa 1992, wyd.II.
4. H. Małysiak, B. Pochopień, E. Wróbel „Procesory arytmetyczne", Seria "Mikrokomputery". WNT Warszawa, 1993.
5. H. Małysiak, B. Pochopień, E. Wróbel „Mikrokomputery klasy IBM PC", Seria „Mikrokomputery", WNT Warszawa, 1992, wyd.II.

23. Total workload required to achieve learning outcomes:

Lp.	Teaching mode	Contact hours / Student workload hours
1	Lecture	30/30
2	Classes	/
3	Laboratory	30/55
4	Project	/
5	Seminar	/
6	Others	5/-
	Total number of hours	65/85

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:

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

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