

(pieczęć wydziału)

COURSE DESCRIPTION

1. Course title: ASSEMBLER PROGRAMMING LANGUAGES		2. Course code: APL		
3. Validity of course description: 2018/2019				
4. Level of studies: first degree				
5. Model of studies: stationary				
6. Field of study: INFORMATICS				
7. Profile of studies: general academic				
8. Programme: ALL				
9. Semester: 4, 5				
10. Faculty teaching the course: Faculty of Automatic Control, Electronics and Computer Science, Institute of Informatics				
11. Course instructor: Ph.D. Eng. Krzysztof Tokarz				
12. Course classification: general				
13. Course status: obligatory				
14. Language: English				
15. Pre-requisite qualifications: Digital Circuits Theory, Arithmetic of Digital Systems, Digital Circuits Design, Fundamentals of Computer Programming				
16. Course objectives: The course aims to provide the basic knowledge and skills in the area of programming using the assembler language. Knowledge obtained gives the ability to make proper choice of programming language and methods to solve specific engineering tasks especially requiring the memory or time optimization. Student will understand details of executing the instructions and programs by the processor, computer functioning and will be able to use other programming languages consciously.				
17. Description of learning outcomes:¹				
Nr	Learning outcomes description	Method of assessment	Teaching methods	Reference code
1	Has the theoretical knowledge about architecture of modern processors and computers, operating systems, memory protection and virtual memory mechanisms.	Exam (sem. 4)	Lecture	K1A_W07, K1A_W10

¹ należy wskazać ok. 5 – 8 efektów kształcenia

2	Student has the knowledge of implementation and optimization algorithms written in assembler language.	Lab report (sem. 4)	Laboratory	K1A_W12
3	Student has the knowledge about the life cycle of modern processors, knows the trends of processors evolution.	Lab report (sem. 4)	Laboratory	K1A_W13, K1A_W14
4	Student can write programs in assembler language and use the API of the operating system.	Project report (sem. 5)	Project	K1A_U23
5	Student can perform systematically the low-level programs verification and testing.	Project report (sem. 5)	Project	K1A_U15
6	Student can create, according to the specification, software written in low-level programming language for simple computer systems.	Project report (sem. 5)	Project	K1A_U26 K1A_U29
18. Teaching modes and hours				
Lecture / BA /MA Seminar / Class / Project / Laboratory:				
30 / 0 / 0 / 0 / 30 / 15				

19 Syllabus description:

Lecture:

- Introduction, Assembler among other programming languages. MASM – Microsoft assembler. Reasons to use assembler in modern software design.
- Architecture and programming model of x86-32 family of processors. Evolution of the Intel processors from 8086 to Core i7. Registers, flags, memory organization, segments, logical, physical, linear address. Memory and peripherals addressing modes. Interrupts, exceptions. Fundamental data types.
- Instruction set of Intel x86-32 family of processors. Format of the instruction, instruction encoding. Main processor instruction set.
- Math coprocessor, floating point instructions, MMX technology. Modern extensions of the instruction set: SSE, SSE2, SSE3, SSSE3, SSE4, AVX.
- Elements of the assembler language: statements, constants, variables, symbolic names, expressions and operators, keywords, directives, pre-defined symbols.
- Creating logical segments, defining variables and constants, data structures, unions, bit records, strings.
- Macros, macro procedures, repeat blocks, procedures, passing parameters to macros and procedures.
- Conditional assembling, interconnections between modules.
- Using assembler in modern programming environments, connecting assembler modules with high level programming languages.

Laboratory:

- Simple assembler program in Visual Studio programming environment.
- Debugging the assembler program, examples of using of the instructions.
- Simple assembler program in Windows operating system.
- Structure of the program with one main window.
- Assembler modules in Visual Studio programming environment.

Project:

Realization of the project written in assembler language with the individually chosen topic. Project should be prepared in Visual Studio programming environment using the mechanism of connecting assembler modules with C language and writing the dynamically linked libraries (dll).

20. Exam: yes (sem. 4)

21. Primary sources:

1. Intel® 64 and IA-32 Architectures Software Developer Manuals available at <https://software.intel.com/en-us/articles/intel-sdm>
2. Wróbel Eugeniusz „Praktyczny kurs asemblera. Wydanie II”, Wydawnictwo Helion, Gliwice 2011.
3. Wróbel Eugeniusz „Programowanie w języku asemblera MASM. Laboratorium”, Wydawnictwo Politechniki Śląskiej, Gliwice 2005.
4. Randal Hyde „Assembler. Sztuka programowania. Wydanie II”, Wydawnictwo Helion, Gliwice 2010.
5. Wróbel Eugeniusz (pod red.) „Assembler. Ćwiczenia praktyczne”, Wydawnictwo Helion, Gliwice 2002.
6. Vlad Pirogov „Assembler. Podręcznik programisty”, Wydawnictwo Helion, Gliwice 2005.

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.

23. Total workload required to achieve learning outcomes

Lp.	Teaching mode	Contact hours / Student workload hours
1	Lecture	30/30
2	Classes	/
3	Laboratory	15/30
4	Project	30/60
5	Seminar	/
6	Other	10/20
	Total number of hours	75/120

24. Total hours: 195**25. Numbers of ECTS:** 6 (3 – sem. 4, 3 – sem. 5)**26. Number of ECTS credits allocated for contact hours:** 3**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)