YLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study / Qualification	Computer science / Engineer
1.7 Form of education	Full time

2. Data about the subject

2.1 Subject name	***Assembly Language Programming***			Subject code	**9**
2.2 Course responsible / lecturer	Assoc. prof. dr. eng. Cebuc Emil - Emil.Cebuc@cs.utcluj.ro				
2.3 Teachers in charge of seminars / Laboratory / project	Assoc. prof. dr. eng. Cebuc Emil - Emil.Cebuc@cs.utcluj.ro Lect. dr. eng. Lișman Dragoș - dragos.lisman@cs.utcluj.ro				
2.4 Year of study	I	2.5 Semester	1	2.6 Type of assessment (E - exam, C - colloquium, V – verification)	E
2.7 Subject category	Formative category: DF – fundamental, DS – speciality, DC – complementary				DS
	Optionality: DOB – mandatory, DOP – optional (alternative), DFA – optional (free choice)				DOB

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars	-	Laboratory	2	Project	-
3.2 Number of hours per semester	56	of which:	Course	28	Seminars	-	Laboratory	28	Project	-
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										10
(b) Supplementary study in the library, online and in the field										17
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										10
(d) Tutoring										4
(e) Exams and tests										3
(f) Other activities:										-

3.4 Total hours of individual study (suma (3.3(a)...3.3(f)))	44
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	None
4.2 Competence	None

5. Requirements (where appropriate)

5.1. For the course	Projector, Blackboard
5.2. For the applications	PC with 32 bit operating system, 1 PC per student, DOSBox

6. Specific competence

6.1 Professional competences	**C2** Designing hardware, software and communication components • **C2.1** Describing the structure and functioning of computational, communication and software components and systems • **C2.2** Explaining the role, interaction and functioning of hardware, software, and communication components • **C2.3** Building the software components of some computing systems using algorithms, design methods, protocols, languages, data structures, and technologies • **C2.4** Evaluating the functional and non-functional characteristics of the computing systems using specific metrics • **C2.5** Implementing hardware, software and communication systems
6.2 Cross competences	N/A

7. Expected Learning Outcomes

Knowledge	Describes, identifies and summarizes elementary concepts and methods regarding programming languages, programming environments and programming techniques, namely: • Basic concepts of an assembly language: syntax, directives, instruction format. x86 ISA architecture: registers, pointers, addressing modes, protected mode. • Instructions: transfer, arithmetic, logical, jump, string, FPU, MMX. • Programming techniques specific to assembly language programming.
Skills	• Selects and explains concepts specific to assembly language programming. • Specifies, elaborates, develops and tests assembly language programs. • Uses programming environments and debugging tools specific to assembly language programming
Responsibilities and autonomy	Has honorable, responsible and ethical behavior, in the spirit of the law to ensure the reputation of the profession.

8. Discipline objective (as results from the *key competences gained*)

8.1 General objective	Knowledge of Microprocessor structure and low-level programming
8.2 Specific objectives	Is able to use various addressing modes, assembly language programming techniques, use specific programming tools

9. Contents

9.1 Lectures	Hours	Teaching methods	Notes
C1. Introduction, data representation	2	PowerPoint presentations, Examples of Program listings, lecture.	
C2. ISAx86 Architecture, addressing modes, CPU registers	2		
C3. x86 Instruction format	2		
C4. MASM x86 directives ALP program prototypes	2		
C5. ISA x86 Instruction set – data transfer, address transfer arithmetic and logical instructions	2		
C6. ISA x86 Instruction set – shift, rotate, flow control instructions	2		
C7. ISA x86 Instruction set – 386, software interrupt, string instructions	2		
C8. Coprocessor structure and operation, data transfer, arithmetic instructions	2		
C9. Coprocessor math functions, misc. instructions	2		
C10. MMX extensions – MMX calculus, MMX instructions	2		
C11. Protected mode operations, memory management, segmentation, privilege levels	2		

C12. System function calls	2		
C13. Multiple module programs	2		
C14. Program optimisation	2		

Bibliography:

1. PPT lecture notes at: ftp.utcluj.ro/pub/users/cemil /ALP
2. D. Gorgan, G. Sebestyen, Proiectarea calculatoarelor", Editura albastra, 2005,
3. R. Hyde R. Hyde, "AoA - The Art of Assembly language", la adresa: webster.cs.ucr.edu/AoA/DOS/pdf/
4. S. Nedevschi, "Microprocesoare", Editura UTCN, 1994

9.2 Applications - Seminars / Laboratory / Project	Hours	Teaching methods	Notes
L1. Information Representation	2	Interactive tutoring, learn by example.	
L2. Tools, ISA x86 Architecture, addressing modes	2		
L3. Addressing Modes and address calculus	2		
L4. Pseudo instruction Usage	2		
L5. ISA x86: Instructions data transfer, arithmetical and logical	2		
L6. ISA x86: Instructions: shift and rotate	2		
L7. ISA x86: Instructions: flow control, other instructions	2		
L8. Real number	2		
L9. Complex operations	2		
L10. Multimedia operations	2		
L11. Program optimisation	2		
L12. System function call	2		
L13. Advanced programming techniques	2		
L14. Colloquium	2		

Bibliography:

1. Art of assembly language, Randall Hyde available at: ftp://ftp.utcluj.ro/pub/users/cemil/asm/
2. Lab Workbook, Emil Cebuc et. All. Available at: ftp://ftp.utcluj.ro/pub/users/cemil/asm/labs/
3. C. Oprisa, A. Hangan, M. Neagu, E. Cebuc, Gh. Sebestyen, „Programare în limbaj de asamblare", Editura UTPRESS Cluj-Napoca, 2018, ISBN 978-606-737-333-2 https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/333-2.pdf

Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

10. Bridging course contents with the expectations of the representatives of the community, professional associations, and employers in the field

Course and lab contents are discussed and compared to similar courses in other universities and with software companies like Bitdefender

11. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Knows microprocessor structure, number representation, x86 basic instruction set, system function calls and assembly program structure	Written examination, Admittance to final exam conditioned by successful lab colloquium	70%
Seminar	-	-	-
Laboratory	Is able to develop a medium size program using specific tools	Weekly mini lab tests Lab Colloquium	30%
Project	-	-	-

Minimum standard of performance:
Can develop a medium size interactive assembly language program using specific tools.
Grade calculus: 30% lab + 70% final exam
Conditions for participating in the Lab Colloquium: ALL lab works have been attended and fulfilled
Conditions for participating in the final exam: Lab Colloquium ≥ 5
Conditions for promotion: final exam ≥ 5

Date of filling in: 20.04.2026	**Responsible**	**Title, First name Last name**	**Signature**
	Course	Assoc.prof.dr.eng. Emil CEBUC	
	Applications	Assoc.prof.dr.eng. Emil CEBUC	
		Lect.dr.eng. Dragoș LIȘMAN	

Date of approval in the department	Head of department, Prof.dr.eng. Rodica Potolea
Date of approval in the Faculty Council	Dean, Prof.dr.eng. Vlad Mureșan