syllabus

1. Information regarding the programme

1.1 Higher education institution	**Babeş-Bolyai University**
1.2 Faculty	**Faculty of Mathematics and Computer Science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Artificial Intelligence**

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)			**Computer Systems Architecture** **Arhitectura Sistemelor de Calcul**				
2.2 Course coordinator			**PhD. Lecturer Coroiu Adriana Mihaela**				
2.3 Seminar coordinator			**PhD. Lecturer Coroiu Adriana Mihaela**				
2.4. Year of study	**2**	2.5 Semester	**3**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Compulsory** **DF**
2.8 Code of the discipline		MLE 5004					

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	0 S 2 LP
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					16
Additional documentation (in libraries, on electronic platforms, field documentation)					15
Preparation for seminars/labs, homework, papers, portfolios and essays					21
Tutorship					14
Evaluations					3
Other activities:					

3.7 Total individual study hours	69
3.8 Total hours per semester	125
3.9 Number of ECTS credits	5

4. Prerequisites (if necessary)

4.1. curriculum	·
4.2. competencies	·

5. Conditions (if necessary)

5.1. for the course	· Video projector
5.2. for the seminar /lab activities	· Laboratory with computers/notebooks

6. Specific competencies acquired

Professional competencies	C1.1 Recognizing and describing specific concepts to calculability, complexity, programming paradigms and modeling of computing and communication systems C1.2 Using specific theories and tools (algorithms, schemes, models, protocols, etc.) for explaining the structure and the functioning of hardware, software and communication systems C2.1 Describing the structure and operation of hardware, software and communication components C2.2 Explaining the role, interaction and operation of hardware, software and communication components
Transversal competencies	CT1 Honourable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation CT3 Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	Knowledge of the computer architecture models, processor functioning, computer information representation usage
7.2 Specific objective of the discipline	- Understanding by the students of the computer architecture models, processor functioning, computer information representation usage - Initiation in assembler language programming, which will assure the comprehension of the microprocessor architecture and functioning - Understanding the impact of the 80x86 processor architecture on Windows functioning and limitations. Awareness of the triade computer architecture – operating systems – programming languages and their interactions as the basic core of Computer Science.

8. Content

8.1 Course	Teaching methods	Remarks
1. Data Representation – Part 1	Exposure, description and conversations.	
2. Data Representation – Part 2	Exposure, description and conversations.	
3. **Computing systems architecture and The 80x86 microprocessor's architecture. The Bus Interface Unit** (BIU) of the 80x86 microprocessor	Exposure, description and conversations.	

4. Assembly language basic elements	Exposure, description and conversations.	
5. Specific instructions for Assembly language	Exposure, description and conversations.	
6. Unsigned Conversions. Specific operations for unsigned numbers.	Exposure, description and conversations.	
7. Signed Conversions. Specific operations for signed numbers.	Exposure, description and conversations.	
8. **The 80x86 microprocessor’s Eflags** register	Exposure, description and conversations.	
9. **Directives** for defining segments, data definition directives.	Exposure, description and conversations.	
10. Overflow concept analysis	Exposure, description and conversations.	
11. Special instructions for strings in Assembly.	Exposure, description and conversations.	
12. Windows Input/Output Function Calls (printf and scanf) and Text files (fopen, fread, fscanf, fprintf, fclose) processing operations	Exposure, description and conversations.	
13. **Multi-module programming** in assembly language	Exposure, description and conversations.	
14. **Review of** theoretical aspects and **additional problems**: integration of the concepts already presented	Description and conversations.	

Bibliography

1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014.

2. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.

3. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005.

4. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003.
(http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html)

5. Boian F.M. Vancea A. Arhitectura calculatoarelor, suport de curs. Facultatea de Matematica si Informatica, Centrul de Formare Continua si Invatamânt la Distanta,. Ed. Centrului de Formare Continua si Invatamânt la Distanta, Cluj, 2002

6. Irvine, K.R., 2015. *Assembly language for x86 processors*.

7. Kusswurm, D., 2014. *Modern X86 Assembly Language Programming*. Springer.

8. Carter, P.A., 2004. *PC Assembly Language*. Github: (http://pacman128.github.io/static/pcasm-book.pdf)

9. Cavanagh, J., 2013. *X86 Assembly Language and C Fundamentals*. CRC Press.

10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer’s manual. *Volume 3B: System programming Guide, Part*, *2*, p.11.
(http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf)

11. Bartlett, Jonathan. "Nasm (Intel) Assembly Language Syntax." In *Learn to Program with Assembly: Foundational Learning for New Programmers*, pp. 271-273. Berkeley, CA: Apress, 2021.

12. Zhirkov, Igor, and Igor Zhirkov. "Assembly Language." *Low-Level Programming: C, Assembly, and Program Execution on Intel® 64 Architecture*, pp 17-38, 2017

8.2 Seminar / laboratory	Teaching methods	Remarks
S1/L1 Conversions between base numbers. Bit. Sign bit. Assembly structure in nasm.	Debate, discovery and problem solving	

S2/L2 Arithmetic expressions and operations for unsigned numbers	Debate, discovery and problem solving	
S3/L3 Arithmetic expressions and operations for signed numbers	Debate, discovery and problem solving	
S4/L4 Bitwise operations in assembly	Debate, discovery and problem solving	
S5/L5 Strings in assembly. Decisional and loops intrctions in asm.	Debate, discovery and problem solving	
S6/L6 Reading and writing (from keyboard, on the screen, from/in file)	Debate, discovery and problem solving	
S7/L7 Multi-module programming in asm.	Debate, discovery and problem solving	

Bibliography

1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014.
2. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
3. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005.
4. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003. (http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html)
5. Boian F.M. Vancea A. Arhitectura calculatoarelor, suport de curs. Facultatea de Matematica si Informatica, Centrul de Formare Continua si Invatamânt la Distanta,. Ed. Centrului de Formare Continua si Invatamânt la Distanta, Cluj, 2002
6. Irvine, K.R., 2015. *Assembly language for x86 processors*.
7. Kusswurm, D., 2014. *Modern X86 Assembly Language Programming*. Springer.
8. Carter, P.A., 2004. *PC Assembly Language*. Github: (http://pacman128.github.io/static/pcasm-book.pdf)
9. Cavanagh, J., 2013. *X86 Assembly Language and C Fundamentals*. CRC Press.
10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer's manual. *Volume 3B: System programming Guide, Part*, *2*, p.11. (http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf)
11. Bartlett, Jonathan. "Nasm (Intel) Assembly Language Syntax." In *Learn to Program with Assembly: Foundational Learning for New Programmers*, pp. 271-273. Berkeley, CA: Apress, 2021.
12. Zhirkov, Igor, and Igor Zhirkov. "Assembly Language." *Low-Level Programming: C, Assembly, and Program Execution on Intel® 64 Architecture*, pp 17-38, 2017

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

The course exists in the studying program of all major universities in Romania and abroad;

The content of the course is considered by the software companies as important for average programming skills

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Testing the basic principles	Written exam	45 %

	of the domain and their interactions		
	Verifying the understanding of the assembly language basic operations and mechanisms	Midterm test (week 7 or week 8)	15 %
10.5 Seminar/lab activities	Application of the 32 bits assembly language principles for problem solving;	Average grade received for the laboratory work	15 %
	Developing and implementing an assembly language code solution for a given problem	Practical exam	15 %
	Evaluating the students activities during the seminaries	Seminar activity	10 %
10.6 Minimum performance standards			
For successfully passing the examination, a student must have at least 5 for the laboratory average, for the written exam, for the practical exam, and minimum 5 as a final grade.			

Date Signature of course coordinator Signature of seminar coordinator

10.04.2023 PhD. Lecturer Coroiu Adriana Mihaela PhD. Lecturer Coroiu Adriana Mihaela

Date of approval Signature of the head of department

.... Prof. dr. Laura Dioşan