



MINISTERUL EDUCAȚIEI
UNIVERSITATEA „AUREL VLAICU” DIN ARAD
310130 Arad, B-dul Revoluției nr. 77, P.O. BOX 2/158 AR
Tel : 0040-257- 283010; fax. 0040-257- 280070
<http://www.uav.ro>; e-mail: rectorat@uav.ro
Operator de date cu caracter personal nr.2929

SYLLABUS

1. Study programme

1.1. Higher education institution	„Aurel Vlaicu” University of Arad
1.2. Faculty	of Exact Sciences
1.3. Department	Department of Mathematics and Computer Science
1.4. Field of study	Informatics
1.5. Study level	2024-2025
1.6. Ciclul de studii	Bachelor
1.7. Study programme / Qualification	Computer Science
1.8. Form of education	Full – Time study

2. Course details

2.1. Name of the course	GIAF2011 Data Structures
2.2. Course coordinator	dr. Bejan Crina-Anina
2.3. Seminar/laboratory/project coordinator	specialist Țerei Carmen
2.4. Study year	1
2.5. Semester	2
2.6. Evaluation type	ES
2.7. Course type	Ob

3. Estimated total time (hours per semester)

3.1. Hours per week	3
3.2. Lecture hours per week	2
3.3. Seminar/laboratory/project hours per week	1
3.4. Total hours per curriculum	42
3.5. Lecture hours per semester	28
3.6. Seminar/laboratory/project hours per semester	14
Time division [hrs]	
3.4.1. Independent study from textbooks, course support, bibliography and notes	30
3.4.2. Additional reading (libraries, specialized electronic platforms and field research)	30
3.4.3. Preparing of seminars/laboratories/projects, homework, papers, portfolios and essays	30
3.4.4. Tutorial coaching	0
3.4.5. Examinations	4
3.4.6. Other activities	0
3.7. Total individual study hours	94
3.8. Total hours per semester	150
3.9. Number of ECTS credits	6

4. Prerequisites (if applicable)

4.1. Curriculum related	
4.2. Competence related	

5. Conditions (if applicable)

5.1. for the lecture	Lecture room, equipped with laptop, video projector and appropriate software.
5.2. for the seminar	
5.3. for the laboratory	Laboratory room, properly equipped: computers, network, Internet connection, appropriate software.
5.4. for the project	

6. Specific educational objectives (competences to be acquired)

6.1. Competențe profesionale	C1. Programming in high-level languages. C2. Development and maintenance of software applications. C3. Using IT tools in an interdisciplinary context. C4. Using the theoretical foundations of computer science and formal models.
6.2. Competențe transversale	CT1. Applying rules of organized and efficient work, demonstrating responsible attitudes towards the educational-scientific field to creatively harness one's potential, while respecting the principles and norms of professional ethics. CT2. Efficiently conducting activities organized within an interdisciplinary group and developing empathetic interpersonal communication skills, as well as the ability to relate to and collaborate with diverse groups. CT3. Using effective methods and techniques for learning, research, and developing the ability to apply knowledge, adapt to the demands of a dynamic society, and communicate in an international language.

7. Course outcomes (resulting from the specific educational objectives to be acquired)

7.1. General outcomes	Students learn the concepts of procedural programming problems and algorithm design and analysis. To develop students' ability to apply correctly the knowledge acquired and to develop their analytical skills.
7.2. Specific outcomes	Students will be able to: - Identify the appropriate algorithm for a given problem; - Design, implement and optimize an algorithm as a solution to a given problem; - Perform complexity calculations for a given algorithm.

8. Outline (if applicable)

8.1 Lecture Outline	Teaching methods	Remarks
Introduction, general notions	interactive exposition, heuristic conversation, exemplification	2 hrs
Linked lists – linked data structures	interactive exposition, heuristic conversation, exemplification	4 hrs
Queues. Stacks.	interactive exposition, heuristic conversation, exemplification	8 hrs
Trees – hierarchical data structures	interactive exposition, heuristic conversation, exemplification	4 hrs
Graphs – relational data structures.	interactive exposition, heuristic conversation, exemplification	4 hrs
Specific Algorithms: Dynamic Programming, Divide and Conquer, Greedy, Backtracking, Branch & Bound	interactive exposition, heuristic conversation, exemplification	4 hrs
Search and sorting methods.	interactive exposition, heuristic conversation, exemplification	2 hrs
8.2 Lecture References 1. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. Introduction to Algorithms. 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN:9780262032933 2; 2. D. Knuth, Arta Programarii Calculatoarelor, Vol.1: Algoritmi Fundamentali, Teora, 2000 3. K.Jamsa, L. Klander, Totul despre C și C++, Manual fundamental de programare în C și C++, Ed. Teora, 2004; 4. V. Iordan, Algoritmi și programare în C, Ed.Eurostampa, 2007 5. D. Galațchi, S. Zoican, R. Zoican, Limbajul C. Structuri de date și algoritmi, Editura POLITEHNICA Press, 2004, ISBN 973-8449-39-1 6. Siddhartha Rao, C++ in One Hour a Day, Sams Teach Yourself, Pearson Education (US), 2016 7. Subrata Saha, Subhodip Mukherjee, Basic Computation and Programming with C, Cambridge University Press, 2017 8. Joseph Bergin, Data Structure Programming: With the Standard Template Library in C++, SPRINGER NEW YORK, 2012		
8.3 Seminar Outline	Teaching methods	Remarks
8.4 Seminar References		
8.5 Laboratory Outline	Teaching methods	Remarks
Introduction to dynamic memory allocation	debate, problem-solving, exercise, application	2 hrs
Linked lists	debate, problem-solving, exercise, application	2 hrs

Queues. Stacks.	debate, problem-solving, exercise, application	2 hrs
Trees	debate, problem-solving, exercise, application	2 hrs
Graphs	debate, problem-solving, exercise, application	2 hrs
Specific Algorithms applications.	debate, problem-solving, exercise, application	2 hrs
Search and sorting methods applications.	debate, problem-solving, exercise, application	2 hrs
8.6 Laboratory References 1. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. Introduction to Algorithms. 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN:9780262032933 2; 2. D. Knuth, Arta Programarii Calculatoarelor, Vol.1: Algoritmi Fundamentali, Teora, 2000 3. K.Jamsa, L. Klander, Totul despre C și C++, Manual fundamental de programare în C și C++, Ed. Teora, 2004; 4. V. Iordan, Algoritmi și programare în C, Ed.Eurostampa, 2007 5. D. Galațchi, S. Zoican, R. Zoican, Limbajul C. Structuri de date și algoritmi, Editura POLITEHNICA Press, 2004, ISBN 973-8449-39-1 6. Siddhartha Rao, C++ in One Hour a Day, Sams Teach Yourself, Pearson Education (US), 2016 7. Subrata Saha, Subhodip Mukherjee, Basic Computation and Programming with C, Cambridge University Press, 2017 8. Joseph Bergin, Data Structure Programming: With the Standard Template Library in C++, SPRINGER NEW YORK, 2012		
8.7 Project Outline	Teaching methods	Remarks
8.8 Project Outline		

9. Correlation of course outline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

The content of the subject is in line with the content of similar subjects in other universities in the country and abroad. In order to better adapt the content of the subject to the requirements of the labour market, meetings were held with employers - representatives of the business environment and with mathematics and computer science teachers from the preuniversity education in Arad.

10. Evaluation / Grading (if applicable)

Activity type	Evaluation criteria	Evaluation methods	Percentage of the final grade
10.1. Lecture	Accuracy and completeness of knowledge. Logical consistency. Degree of assimilation of specialist language.	Oral assessment (final in the exam session): -Presentation of a final project - Free student presentation - Evaluation conversation - Oral questionnaire.	50%
10.2. Seminar			
10.3. Laboratory	Ability to operate with assimilated knowledge. Ability to apply in practice.	Oral assessment (final in the examination session): -Completion and presentation of the final project	30%
10.4. Project	Homeworks.	Oral evaluation.	20%
10.5 Minimal performance standard Learning fundamental concepts, using specialist language, making a simple application.			

Course coordinator
Conf. univ. dr. Crina Anina Bejan

Seminar/laboratory/project coordinator
specialist Carmen Țerei

Head of the Department
Lector Popa Lorena

Dean
Prof.univ.dr. Sorin Florin Nădăban