



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	GIAF2010 Fundamental Algorithms
2.2. Course coordinator	dr. Valeriu BEIU
2.3. Seminar/laboratory/project coordinator	IT specialist Oana-Roxana IACOB
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	4
3.2. Lecture hours per week	2
3.3. Seminar/laboratory/project hours per week	2
3.4. Total hours per curriculum	56
3.5. Lecture hours per semester	28
3.6. Seminar/laboratory/project hours per semester	28
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)	20
3.4.3. Preparing of seminars/laboratories/projects, homework, papers, portfolios and essays	30
3.4.4. Tutorial coaching	10
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	Room equipped with blackboard/interactive-board, video projector and internet connection/recording (for presentations)
5.2. for the seminar	Room equipped with blackboard/interactive-board, video projector, internet connection/recording, and software tools (for demonstrations)
5.3. for the laboratory	Room equipped with blackboard/interactive-board, internet connection/recording, networked computers and software tools (for training and simulations)
5.4. for the project	

6. Specific educational objectives (competences to be acquired)

6.1. Competențe profesionale	C1. Programming in high level programming languages C3. Using computer tools in interdisciplinary context C4. Using the theoretical bases of computers and formal models
6.2. Competențe transversale	CT1. Applying the rules of organized and efficient work, of responsible attitudes towards teaching-scientific field, to value the own creative potential, while respecting the principles and norms of professional ethics. CT3. Using of efficient methods and techniques for learning, informing, research and development of the capacity to value knowledge, adapting to the requirements of a dynamic society and communicating in English and in an internationally widespread language.

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

7.1. General outcomes	Students will acquire fundamental concepts pertaining to the problems of design and analysis of algorithms as well as procedural programming. Develop students' abilities of correctly applying the knowledge acquired in sync with developing their analytical skills.
7.2. Specific outcomes	Students should be able to: <ul style="list-style-type: none"> • Identify an efficient algorithm for a given problem • Design / implement / optimize an algorithm solving a given problem • Estimate the complexity (time and space) of a given algorithm

8. Outline (if applicable)

8.1 Lecture Outline	Teaching methods	Remarks
1.- Introduction; marching of AI (chess, go, etc.) 2.- A bit of history: al-Khwārizmī, Ada Lovelace, Babbage; Shannon, Turing; maze and Shannon's mouse; towers of Hanoi (algorithmic complexity) 3.- Algorithmic thinking; number representations; addition (alternatives) 4.- Data representation and data analysis (ethical issues) 5.- Limits; greatest common divisor; sorting (Google and others) 6.- Recursion / recurrences; space- and time-complexity 7.- Number multiplication; matrix multiplication; systolic multiplication; parallel and quantum algorithms	<ul style="list-style-type: none"> • Presentation • Lecture using video projector, internet • Interactive discussions / conversations • Reading (from the web) • Comparative analyses • Examples • Brainstorming • Modeling 	2 + 4 + 2 + 2 + 10 + 4 + 4 hours
8.2 Lecture References <ol style="list-style-type: none"> 1. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to Algorithms (3rd ed.). Cambridge: MIT Press, 2009. ISBN: 978-0262033848 2. D. Knuth. The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd ed.). Boston: Addison-Wesley, 1997. ISBN: 978-0201896831 3. C.H. Papadimitriou. Computational Complexity (1st ed.). Boston: Addison Wesley, 1994. ISBN: 0201530821 4. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. The Design and Analysis of Computer Algorithms. Boston: Addison-Wesley, 1974. ISBN: 0201000296 5. I. Wegener. The Complexity of Boolean Functions. Wiley-Teubner, 1987. ISBN: 978-0-471-91555-3. https://eccc.weizmann.ac.il/resources/pdf/cobf.pdf 6. M.R. Garey, and D.S. Johnson. Computers and Intractability. W.H. Freeman and Co., 1979. ISBN: 0-7167-1045-5 7. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge: MIT Press, Nov. 2016, ISBN: 978-0262035613. http://www.deeplearningbook.org/ 8. V. Beiu, M. Calame, G. Cuniberti, C. Gamrat, Z. Konkoli, D. Vuillaume, G. Wendin, and S. Yitzchaik, Aspects of Computing with Locally Connected Networks, AIP International Conference on Numerical Analysis and Applied Mathematics (ICNAAM'12), Kos, Greece, Sept. 19-25, 2012, pp. 1875–1879. https://doi.org/10.1063/1.4756547 9. V. Beiu, and L. Dăuș, Reliability Bounds for Two Dimensional Consecutive Systems, Nano Communication Networks (Special Issue on Biological Information and Communication Technology), vol. 6, no. 3, Sept. 2015, pp. 145–152. https://doi.org/10.1016/j.nancom.2015.04.003 10. L. Dăuș, and V. Beiu, Lower and Upper Reliability Bounds for Consecutive-k-out-of-n:F Systems, IEEE Transactions on Reliability, vol. 64, no. 3, Sept. 2015, pp. 1128–1135. https://doi.org/10.1109/TR.2015.2417527 11. S.R. Cowell, V. Beiu, L. Dăuș, and P. Poulin, On Hammock Networks – Sixty Years After, IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS'17), Palma de Mallorca, Spain, Apr. 4-6, 2017, art. 7929871 (pp. 1–6). https://doi.org/10.1109/DTIS.2017.7929871 12. L. Dăuș, V. Beiu, S.R. Cowell, and P. Poulin, Brick-Wall Lattice Paths and Applications, Tech. Rep. arXiv [math.CO], 14 Apr. 2018, pp. 1–16. https://arxiv.org/abs/1804.05277 13. S.R. Cowell, M. Nagy, and V. Beiu, A Proof of a Generic Fibonacci Identity from Wolfram's MathWorld, Theory and Applications of Mathematics & Computer Science, vol. 8, no.1, Apr. 2018, pp. 60–63. 14. M. Nagy, S.R. Cowell, and V. Beiu, Are 3D Fibonacci Spirals for Real? — From Science to Arts and Back to Science, IEEE International Conference on Computers Communications and Control (ICCCC'18), Băile Felix/Oradea, Romania, May 08-12, 2018, pp. 91–96. https://doi.org/10.1109/ICCCC.2018.8390443 15. S.R. Cowell, V. Beiu, L. Dăuș, and P. Poulin, On the Exact Reliability Enhancements of Small Hammock Networks, IEEE Access, vol. 6, no. 1, Oct. 2018, pp. 25411–25426. https://doi.org/10.1109/ACCESS.2018.2828036 16. M. Nagy, S.R. Cowell, and V. Beiu, Survey of Cubic Fibonacci Identities – When Cuboids Carry Weight, Intl. J. Comp. Comm. & Ctrl., vol. 17, no. 2, art. 4616 (pp. 1–20), Apr. 2022. https://doi.org/10.15837/ijccc.2022.2.4616 		

8.3 Seminar Outline	Teaching methods	Remarks
1.- Introduction to programming (structural, procedural); compiling 2.- Data types; variables; operators; expressions 3.- I/O functions; decisions; control; selection (from multiple alternatives) 4.- Instructions: repetitive (loops); control; jumps; return 5.- Functions; recursion 6.- Working with arrays (matrices); searching, ordering, ranking, sorting, classification	<ul style="list-style-type: none"> • Examples • Exercises • Applications • Case studies • Problem formulation • Debate • Analyses and comparisons 	2 + 2 + 4 + 8 + 4 + 8 hours
8.4 Seminar References <ol style="list-style-type: none"> 1. V. Beiu, Neural Networks Using Threshold Gates – A Complexity Analysis of Their Area- and Time-Efficient VLSI Implementations, PhD dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, U.D.C. 621.3.04977: 681.3*C13 (x-27-151779-3), pp. 1–222, May 1994. 2. V. Beiu, Adder Having Reduced Number of Internal Layers and Method of Operation Thereof, US 6,438,572, Aug. 20, 2002, pp. 1–11. https://patents.google.com/patent/US6438572/ [Also as WO/2001/023992 and AU40251/01] 3. V. Beiu, Adder Circuits Employing Logic Gates Having Discrete Weighted Inputs and a Method of Operation Therewith, US 6,502,120, Dec. 31, 2002, pp. 1–13. https://patents.google.com/patent/US6502120/ 4. V. Beiu, Microprocessor and a Digital Signal Processor Including Adder and Multiplier Circuits Employing Logic Gates Having Discrete and Weighted Inputs, US 6,516,331, Feb. 4, 2003, pp. 1–14. https://patents.google.com/patent/US6516331/ 5. https://app.codility.com/programmers/ 		
8.5 Laboratory Outline	Teaching methods	Remarks
8.6 Laboratory References		
8.7 Project Outline	Teaching methods	Remarks
8.8 Project References		

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

This course is taught in similar programs at many universities, both in Romania and abroad. For a better matching with the demands of the labor market, meetings with employers' representatives and specialty teachers from the pre-university education system have been organized. Using English brings and added value to the program, raising the graduates chance to be hired by multinational companies (both abroad and in Romania).

10. Evaluation / Grading (if applicable)

Activity type	Evaluation criteria	Evaluation methods	Percentage of the final grade
10.1. Lecture	<ul style="list-style-type: none"> • Level of mastering the domain-specific vocabulary • Logical consistency • Extent of correctness and completeness of knowledge • Responsibility • Commitment • Resolution 	<ul style="list-style-type: none"> • Q&A and other contributions during lectures • Conversations • Final evaluation (exams) • Active participation 	30% 10%
10.2. Seminar	<ul style="list-style-type: none"> • Ability to use the acquired knowledge • Ability to apply theoretical knowledge to practical cases • Responsibility • Commitment • Resolution 	<ul style="list-style-type: none"> • Exams • Reports and presentations • Homework • Active participation 	30% 10% 20%
10.3. Laboratory			
10.4. Project			
10.5 Minimal performance standard Proper mastering of basic concepts, able to use the domain-specific vocabulary, ability to design and implement simple applications.			

Course coordinator
Prof. dr. Valeriu BEIU

Seminar/laboratory/
project coordinator
Specialist IT
Oana-Roxana IACOB

Head of the Department
Lecturer dr. Lorena POPA

Dean
Prof. dr. Sorin NĂDĂBAN