

I047	Competitive Programming	L	S	E	ECTS 6
		2	0	2	

Course objectives. The main course objective is to develop skills that are necessary for student participations in programming contests. During the semester students actively solve very complex programming problems that appeared on past programming contests. Beside correctness and effectiveness, the speed of problem solving in the contest environment is emphasized.

Course prerequisites. Introduction to Computer Science. Object-Oriented Programming.

Syllabus.

1. Introduction. Overviews and Rules of Programming Contests. Writing of programming code in contest environment. Problem identification. Algorithmic analysis. Implementation. Input instance Generation. Code evaluation. Team work. Online judge platforms.
2. Data Structures and Libraries.
3. Algorithmic techniques.
4. Graph algorithms.
5. Algorithms in Combinatorics, Number Theory, and Probability.
6. String Processing.
7. Computational Geometry algorithms.
8. Advanced Topics.

EXPECTED LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	Fast and accurate algorithmic analysis of complex programming problems.
2.	Fast coding and debugging in contest environment.
3.	Space and time efficiency analysis. Correctness analysis.
4.	Efficient application of different data structures and algorithms in contest environment.
5.	Programming contests attendance.

COUPLING OF THE EXPECTED LEARNING OUTCOMES, TEACHING PROCESS ORGANIZATION AND THE EVALUATION OF THE TEACHING OUTCOMES

TEACHING PROCESS ORGANIZATION	ECTS	EXPECTED LEARNING OUTCOMES **	STUDENT ACTIVITY*	EVALUATION METHOD	SCORE	
					min	max
Lecture attendance	2	1-4	Class attendance, discussion, solving the problems individually and in a team	Lists with signatures, observing the activity during the lectures	20	40

Simulated competitions	4	5	Attending real and simulated programming competitions	Evaluating the results	30	60
TOTAL	6				50	100

Teaching methods and student assessment.

During lectures, the process of problem solving in contest environment is presented to students. During exercises, students individually solve programming problems according the methods that are presented in the lecture session. The main goal is to correctly solve programming problems, as many as possible. Students develop skills of team work during the practice contests. Final grade is formed according to the total number of individually solved problems during classes and the total number of solved problems in practice and/or real contests.

Can the course be taught in English: Yes

Basic literature:

1. S. Halim, F. Halim, Competitive Programming 3

Recommended literature:

1. D. Kusalić, Napredno programiranje i algoritmi u C-u i C++-u, Element, Zagreb, 2014.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 3Ed, MIT Press, 2009.
3. J. Šribar, B. Motik: Demistificirani C++, 4. dopunjeno izdanje usklađeno sa standardom C++11/C++14, Element, Zagreb, 2014.
4. M. T. Goodrich, R. Tamassia, D. M. Mount, Data Structures and Algorithms in C++, Wiley, 2010.
5. A. Drozdek, Data Structures and Algorithms in C++, Cengage Learning, 2012.
6. R. Sedgewick, K. Wayne, Algorithms, Addison-Wesley Professional, 2011.
7. M. J. Atallah, Algorithms and Theory of Computation Handbook, CRC Press, 1998.
8. R. Sedgewick, Algorithms in C++, Parts 1-4 Fundamentals, Data Structure, Sorting, Searching, Third Edition, 1998.