

Study Programme: Computer Science		
Course Unit Title: Introduction to Programming		
Course Unit Code: CS101		
Name of Lecturer(s): Miloš Radovanović		
Type and Level of Studies: Bachelor Academic Degree		
Course Status (compulsory/elective): Compulsory		
Semester (winter/summer): Winter		
Language of instruction: Serbian (primary), English (secondary)		
Mode of course unit delivery (face-to-face/distance learning): Face-to-face		
Number of ECTS Allocated: 8		
Prerequisites: None		
Course Aims: Enabling the student to understand the concepts of (computer) programming, problem analysis and their implementation in a concrete programming language using the procedural programming style.		
Learning Outcomes: <i>Minimum:</i> At the end of the course it is expected from a successful student to demonstrate understanding of the concepts of (computer) programming, ability to understand problems and the implementation of solutions in a concrete programming language, with the use of available libraries. <i>Desirable:</i> At the end of the course it is expected from a successful student to demonstrate a deep understanding of the concepts of (computer) programming, ability to understand and analyze problems and implement solutions using the procedural programming style.		
Syllabus: <i>Theory</i> Computers, programs, programming languages, Java. Program elements and structure. Primitive data types, statements, expressions. Control and iteration statements. References, referential types. Classes, arrays, enums. Class members: fields, methods. Introduction to algorithms. Recursion. Iterative and recursive approaches. Introduction to abstract data types. Example implementation of an abstract data type, using arrays, in procedural and object-oriented styles. <i>Practice</i> Practicing the understanding of basic principles of the procedural programming style. Practicing control and iteration statements, as well as primitive, referential, and abstract data types. Practical implementation of algorithms using an appropriate editor and compiler.		
Suggested Reading: 1. D. J. Eck. Introduction to Programming Using Java. 7th edition, 2014. http://math.hws.edu/javanotes/ 2. J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley. The Java Language Specification. Oracle America, Inc., Java SE 8 edition, 2015. 3. R. Gallardo, S. Hommel, S. Kannan, J. Gordon, S. B. Zakhour. The Java Tutorial: A Short Course on the Basics. Addison-Wesley, 6th edition, 2015.		
Weekly Contact Hours: 5	Lectures: 2	Practical work: 3
Teaching Methods: Lectures are organized using classic teaching methods with use of a projector. Principles of procedural programming are		

explained and illustrated with appropriate examples in a concrete programming language (currently, Java). Exercises are performed with classic teaching methods, where principles of procedural programming are practiced through illustrative examples. Computers are used to implement solutions to problems in an appropriate environment (editor+compiler). Students' knowledge acquired at lectures and exercises is evaluated through solution of four practical problems (compulsory) and two written tests (elective). At the oral part of the exam, by answering questions students demonstrate deeper understanding of analysis and implementation of problem solutions using the procedural programming style.

Knowledge Assessment (maximum of 100 points):

Pre-exam obligations	points	Final exam	points
Practical exercises – individual problems	60	Oral examination (obligatory)	20-40
		Tests (elective)	0-20

The methods of knowledge assessment may differ; the table presents only some of the options: written exam, oral exam, project presentation, seminars, etc.