

Study Programme: Applied Mathematics – Data Science			
Course Unit Title: Advanced programming for mathematicians			
Course Unit Code: MDS30			
Name of Lecturer(s): Srđan M. Škrbić			
Type and Level of Studies: Master Academic Degree			
Course Status (compulsory/elective): elective			
Semester (winter/summer): Winter			
Language of instruction: English			
Mode of course unit delivery (face-to-face/distance learning): Face-to-face			
Number of ECTS Allocated: 5			
Prerequisites: None			
Course Aims: This is an introductory course for C++. It is intended for students of mathematics with little programming experience. The goal is to study main constructions of C++, including objectoriented programming and to gain basic knowledge of the usage of this programming language for scientific purposes.			
Learning Outcomes: <i>Minimal:</i> At the end of the course, it is expected that students understand basic concepts of the C++ language, including object-oriented programming. It is also expected that students know how to apply this knowledge in practical programming and science. <i>Desirable:</i> At the end of the course, it is expected that a successful student shows ability to identify optimal way to solve specific given problem using C++. It is also expected that a successful student has active knowledge of all advanced concepts of this language.			
Syllabus: <i>Theory</i> At the beginning of the course, an introduction to basic constructions of C++ is given – program structure, data types, constants, operators and flow control. In the continuation, functions, arrays, strings and pointers are presented. The second part of the course is an introduction to objectoriented programming using C++. <i>Practice</i> Inside exercise classes, concepts covered by lectures are practiced through examples motivated by science applications. The accent is on usage of object-oriented programming and its applications in science.			
Required Reading: 1. Bjarne Stroustrup, The C++ Programming Language, Addison-Wesley, 2000. 2. Scott Meyers, Effective C++, Addison-Wesley, 2005. 3. Bruce Eckel, Thinking in C++, Prentice Hall, 2000. 4. William Press, Saul Teukolsky, William Vetterling, Brian Flannery, Numerical Recipes: The Art of Scientific Computing, Third Edition in C++, 2007.			
Weekly Contact Hours:	Lectures: 2	Practical work: 2	
Teaching Methods: During theoretical classes classical methods of teaching with the use of a projector are used to present stated topics. On practical classes, classical methods of teaching with the use of a projector and computers with appropriate software installed are used to practically master the skills of usage of suggested tools. A premise for successful practical classes is the existence of enough computers so that every student may work individually.			
Knowledge Assessment (maximum of 100 points):			
Pre-exam obligations	points	Final exam	points
Active class participation		written exam	

Practical work		oral exam	50
Preliminary exam(s)		
Colloquia	50		
Seminar(s)			
The methods of knowledge assessment may differ; the table presents only some of the options: written exam, oral exam, project presentation, seminars, etc.			