

Study program: Information Technology			
Course title: INTRODUCTION TO PROGRAMMING			
Teacher(s): Nebojša Lj. Stanković, Marija D. Blagojević			
Course status: mandatory			
Number of ECTS credits: 6			
Prerequisite courses: none			
Course objectives Learning the fundamental ideas and coding techniques required to create high-quality programs. Developing algorithmic schemes and writing code through lab exercises to practice fundamental programming approaches. Using the C programming language to create, document, test, and implement a well-structured computer program.			
Learning outcomes By the end of the course, the student will be able to: explain the benefits of high-level programming languages, the programming process, and compilation; describe and use software tools in the programming process; use an integrated development environment to design algorithmic schemes to solve straightforward problems; design, implement, debug, test, identify, and debug programming code using the fundamentals of the C programming language.			
Content of the course <i>Theoretical teaching</i> Generations of programming languages: machine, symbolic (assembly) languages, high-level symbolic languages. Programming in machine and symbolic machine language. Algorithms and issue solving: problem solving techniques, the function of algorithms in problem solving, algorithm implementation techniques, algorithm debugging techniques, algorithmic structures, and more. Fundamental programming constructs: syntax and semantics of higher-level languages; variables, types, expressions and assignment, data input/output; conditional and iterative control structures. Functions and passing parameters; structured decomposition. Arrays, characters, strings and functions for processing them. Data structures. Files (binary and text). <i>Practical teaching</i> Realization of the problem in practice. Development of algorithmic schemes. Creating programs in the C programming language (program flow control, functions, arrays, strings, structures, files). Writing and executing programs. Using the compiler and putting the program to the test.			
Literature [1] Sebesta, R., <i>Concepts of programming languages</i> , Pearson Education Limited, 2022, ISBN 9781292436821 [2] Kurhekar, A.S., <i>An Introduction to C Programming</i> , Independently Published, 2016, ISBN 9781292100555 [3] Cormen, T.H., <i>Introduction to Algorithms</i> , fourth edition, The MIT Press, 2022, ISBN 9780262046305 [4] Zingaro, D., <i>Algorithmic Thinking: A Problem-Based Introduction</i> , No Starch Press, 2021, ISBN 9781718500815 [5] Kraus, L., <i>Programski jezik C sa rešanim zadacima – 10. izdanje</i> , Akademska misao, Beograd, 2020, ISBN 978-86-7466-831-3 [6] Kraus, L., <i>Rešeni zadaci iz programskog jezika C</i> , Akademska misao, Beograd, 2020, 978-86-7466-856-6 [7] Urošević, V., Ristić, O., Blagojević, M., Savić, B., <i>Programski jezik C: zbirka zadataka</i> , Fakultet tehničkih nauka Čačak, Čačak, 2013, ISBN 978-86-7776-151-6 [8] Zimmerman, C., <i>Pravila programiranja</i> , 2023, Mikro knjiga, 2023, ISBN 978-86-7555-467-7			
Number of active teaching classes: 6		Theoretical teaching: 2	Practical teaching: 4
Teaching methods Interactive teaching (teaching methods: popular lecture, discussion, practical workmethods, workshops); activated forms of learning: verbal meaningful receptive learning, discovery learning, cooperative learning, practical learning, independent student work.			
Evaluation of knowledge (maximum number of points 100)			
Pre-exam obligations	Points	Final exam	Points
Activities during teaching process	10	Final exam (written):	20
Tests/homework	30	Final exam (oral):	10
Colloquiums	30		