

| | | | | | | | |
|---|--|------------------|--------------------------|-------------------------|----|-----------|-----|
| Program | Level | | Short cycle | | | | |
| | Name of the program | | Information Technologies | | | | |
| COURSE | | | | | | | |
| Course title | Comparative Analysis of Programming Languages | | | | | | |
| Course code | Semester | Course status | ECTS | Contact hours (L+AE+LE) | | | |
| IT 260 | IV | Mandatory course | 5 | 3+0+2 | | | |
| Lecturer | | | | | | | |
| Course Goals | The module is an advanced programming course. The goals of the module are to get acquainted with several programming languages that students have not had the opportunity to encounter in other modules. The course also considers and compares different programming paradigms such as procedural, object-oriented, functional and logical programming. | | | | | | |
| Learning Outcomes | <p>Upon completion of the module, students will be able to:</p> <ul style="list-style-type: none"> - Understand what functional, object-oriented and logical programs mean - Learn new programming languages quickly - Recognize the characteristic concepts of different programming languages - Determine the strength and elegance of programming languages and their parts - Use new programming languages - Use already known languages more skillfully | | | | | | |
| COURSE CONTENT | | | | | | | |
| <ul style="list-style-type: none"> - Functional programming (change avoidance; use of recursion and higher order functions, closures, anonymous functions) - Algebraic data types and pattern matching - Object oriented programming - Abstractions - Syntax and semantics - Static and dynamic types, type safety - Polymorphism and generic programming - Logical programming | | | | | | | |
| LITERATURE | | | | | | | |
| <p>[1] R. W. Sebesta, Concepts of Programming Languages, 11th Edition, 2015 [2] M. L. Scott, Programming Language Pragmatics, 4th Edition, 2015. [3] A. Tucker, R. Noonan, Programming Languages: Principles and Paradigms, 2nd Edition, 2006. [4] H. Abelson, G. J. Sussman, J. Sussman, Structure and Interpretation of Computer Programs, 2nd Edition, 1996.</p> | | | | | | | |
| STUDENT WORKLOAD (hours in a semester) | | | | | | | |
| Lectures | 45 | Exercises | 30 | Individual work | 50 | T o t a l | 125 |
| GRADING | | | | REMARKS | | | |
| Criterion | Maximum points | Minimum points | | | | | |
| Laboratory assignments | 30 | | | | | | |
| Tests | 30 | | | | | | |
| Final exam | 40 | | | | | | |
| T o t a l | 100 | 55 | | | | | |