**Table 5.1** Course specification to doctoral study programs

|  |
| --- |
| **Course name: Scientific programming** |
| **Teacher or teachers:** [**Gocić Lj. Milan**](../P%209.3%20Knjiga%20Nastavnika%20DOS%20He/53.%20Milan%20Lj.%20Gocic%2C%20docent.xlsx) |
| **Course status:** Elective |
| **Number of ECTS:** 10 |
| **Precondition courses:** None |
| **Educational goal**The aim of this course is to give students an introduction to the uses of computer languages in the analysis of contemporary scientific problems and how to set up a scientific environment.  They will be able to acquire knowledge on scientific computing to deals with solving scientific problems with the help of computers and to obtain results more quickly and accurately.  |
| **Educational outcomes** Students will learn the basic usage of each language, common types of problems encountered, and techniques for solving a variety of problems encountered in contemporary research: examination of data with visualization techniques, numerical analysis, and methods of dissemination and verification. This course introduces programming languages and techniques used by scientists: C#, MATLAB and Mathematica.  |
| **Course content**Scientific programming and scientific computing. Problem formulation, algorithm development, algorithm implementation, and algorithm verification. Programme structure and documentation. Scientific programming languages. Structured programming and structured code. Scientific programming in C#: Representation of data through arrays and data structures; Function calls, argument passing and scoping rules; Encapsulation and inheritance. Unit testing. Specialist systems. Mathematica: structure of Mathematica; symbols, exact numbers, and machine numbers; lists, vectors, and matrices; numerical calculations, symbolic calculations, and graphics. Importing and exporting information. MATLAB: MATLAB syntax, workspace, variables; Script M-files, IO, control flow, debugging, and profiling tools; Object-oriented programming; MATLAB applications, polynomials, interpolation, integration, differentiation, Ordinary differential equation (ODE) solutions; Graphics, 2-D, 3-D, Graphical User Interface.  |
| **Literature**1. M. Gocić, Programski jezik C#: pitanja, odgovori i rešeni zadaci, Mikro knjiga, Beograd, 20132. W. Stephen, The Mathematica Book, 5th ed., Wolfram Media, 20033. Cleve B. Moler, Numerical computing with Matlab, Society for Industrial and Applied Mathematics, 20044. Andrew Troelsen, Pro C# 5.0 and the .NET 4.5 Framework, Apress, 2012 |
| **Number of active teaching classes (weekly)** | Lectures: 4 | Study research work: 0 |
| **Teaching methods**Lectures. Consultations and interactive work with the students. Lectures are organized in combined form. The presentation of the theoretical part is followed by the corresponding examples. |
| **Knowledge evaluation (maximum 100 points)****Pre-examination obligations Points Final exam Points**Lecture attendance **10**  Oral part of the exam **30**Project task **30**Term paper **30**  |