Course PYTHON LANGUAGE FOR SCIENTIFIC RESEARCH

Basic Information

This is a course, which contributes to MSc award in Mathematics

Duration	**72 hours (2 ECTS)**
Starting date	September, 1st
Study credits	2 ECTS credits
Language of instruction	English level B1(European Framework of Reference of Communicative Skills)
Academic requirements	– BSc degree in Mathematics, Physics, Computer Science, Engineering or equivalent (a copy of your diplomas from previous university studies and transcripts of completes courses and grades), – Skype interview

Course Description

The Python Programming Language for Scientific Computing is an optional discipline for exploring existing Python libraries. The main goal of the course is to use the Python language to solve applied scientific problems. The discipline discusses the use of basic Python language packages for solving various scientific problems based on the integration of programming with applied computing in mathematics, physics, biology and economics.

Special Features of the Course

Python is one of the easiest and most efficient programming languages to date. It can be used both for the initial steps in programming, and for solving serious scientific problems. The advantage of the Python language is a large number of libraries ready for use in various fields: data analysis and machine learning, scientific computing, visualization, network programming, etc. due to which its application in scientific work makes it easy to solve a wide range of problems.

Course Aim

- Explore the main libraries of the Python language used in scientific work.
- To develop skills for solving applied problems using the Python language.

Course Objectives

- Explore matplotlib data visualization library.
- Explore the numpy compute library.
- Explore the sympy symbol computing library.
- Explore scipy science computing library.

Learning Outcomes of the Course

By the end of the course, students will be able to:

- use Python graphical functions;
- apply linear algebra techniques in Python;
- perform symbolic conversions in Python;
- perform numerical integration and differentiation in Python.

Course (module) Structure

Learning Activities	Hours
Practice sessions / Seminars,	18
Self-study Assignments	54
Total study hours	**72**

Course Outline

Week	Practice session / Assignments	Hours[1]
1-3	– Data visualization – 2d matplotlib graphics – 3d graphics matplotlib – Home assignment No 1	12
4-8	– Computational math in Python – Module random – Linalg module – Fft module – Polynomial module – Home assignment No 2	20

[1] Hours designed for Classroom sessions, Web-sessions, Home Assignments etc.

9-13	– Sympy Character Computing Library – Basics of symbolic computing – Algebraic calculations – Mathematical analysis in Python – Symbolic and numerical mathematics – Home assignment No 3	20
14-18	– Scipy math library – Numerical integration – Numerical differentiation – Home assignment No 4	20

Assessment

Grade policy for home assignments is:

- A (excellent work) 91–100 points
- B (above average work) 81–90 points
- C (average work) 71–80 points
- D (below average work) 50–70 points
- F (failed work) < 50 points

Final standings are conducted in the form of an interview on sections of the discipline
The final grade includes grade for homework and interview in a ratio of 50:50.

Lecturer(s) and Tutors, Contact Information

Roman ESIN
Senior Lecturer at Institute of Space and Information Technologies, Siberian Federal University
(room 311) 26-1, Kirensky st, Krasnoyarsk, Russia
Tel: +7 391 291 2790,
resin@sfu-kras.ru, http://ikit.sfu-kras.ru/e/207

Core Reading

All the books are available on-line in Elsevier at SFU library.

TJ O'Connor, Violent Python, 1st Edition, A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers, Paperback ISBN: 9781597499576, eBook ISBN: 9781597499644, Imprint: Syngress, Published Date: 8th November 2012, Page Count: 288