Course ID 018248

Programming with Python

MATH 20622

Credit rating 10

ECTS credits 5

Unit coordinator: Stefan Guettel

Semester 2

School of Mathematics

Undergraduate

Level 2

FHEQ level ' Middle part of Bachelors'

Marketing course unit overview

The course will focus on teaching basic programming mechanisms in Python and on writing programs based on basic mathematical algorithms, as well as creating algorithms from various problems' descriptions.
Most of the Python-specifc constructs will be avoided, to make students familiar with the concepts that they can easily apply to other programming environments they may encounter.
No prior computer programming knowledge is assumed.

Course unit overview

The course will focus on teaching basic programming mechanisms in Python and on writing programs based on basic mathematical algorithms, as well as creating algorithms from various problems' descriptions.
Most of the Python-specifc constructs will be avoided, to make students familiar with the concepts that they can easily apply to other programming environments they may encounter.
No prior computer programming knowledge is assumed.

Aims

To introduce students to the \algorithmic way of thinking" and to the basic programming concepts, laying strong foundations for computer-aided problem solving applicable in their later studies, as well as for some programming-related problems they may encounter in the future.

Learning outcomes

On successful completion of this course, students will be able to translate existing algorithms to Python. Students will also be capable of presenting and solving mathematical problems in a form understandable by a computer, with the practical work done in Python, but applicable to any other similar language.

Syllabus

1. Introduction. Programs as computers' \cookbooks". Compilers vs. pseudo-compilers. Python versions (no details, just the awareness that there are two which are somewhat incompatible) and IDEs (for Linux, Mac, Windows). Simple program (writing, running, testing in the chosen IDE (probably Anaconda)).

2. Basic input and output, variables, basic types in Python (numbers, strings; mention types in typed languages), value assignments, simple expressions (standard arithmetic operations, real and integer division, division remainder,. . .). Boolean expressions, true and false. Comments and their importance.
3. Branching (if, if...elif...else). The concept of \doing nothing" (pass).
4. Loops (for and range, while). Nesting loops. Flow control (continue, break).
5. Recap of the above: simple algorithms (Euclid's algorithm, \toying" with numbers' digits, divisibility, etc).
6. Subroutines (de ning, arguments, return values, documenting), reusing previously presented algorithms.
7. Lists (including multidimensional ones): concept, basic usage and operations (adding items, traversing, removing items), passing them to and from subroutines. More complex operations (sorting and searching).
8. Concept of complexity (when is one algorithm \better" than some other). Examples of simple algorithms highly improved by reduction of complexity.
9. Modules (loading them, introduction to some of the common ones). Brief introduction to the Python package ecosystem (NumPy, SciPy, SymPy, etc.).
10. Basics of testing and debugging. File I/O.

Assessment methods

Three coursework programming projects and possibly one or two in-class mini-tests.

Requisites

MATH10101	Foundations of Pure Mathematics A	Pre-Requisite	Compulsory
MATH10111	Foundations of Pure Mathematics B	Pre-Requisite	Compulsory

Students should have taken (MATH10101 OR MATH10111)

Available as free choice? N

Scheduled activity hours

Lectures	11
Practical classes & workshops	22

Independent study hours 67 hours

Version Nbr 005.0.0