Faculty of Science

NUMA03, Programming Concepts with Python, 7.5 credits

Programmeringskoncept med Python, 7,5 högskolepoäng

First Cycle / Grundnivå

Details of approval

The syllabus was approved by The Education Board of Faculty of Science on 2026-06-05. The syllabus comes into effect 2026-06-05 and is valid from the spring semester 2027.

General information

The course is a compulsory first-cycle course for a Bachelor of Science degree in Physics and is included as a compulsory course in the Medical Radiation Physics Programme. The course is also offered as free-standing course.

Language of instruction: English

Main field of study	*Specialisation*
Mathematics with specialization in Numerical Analysis	G1F, First cycle, has less than 60 credits in first-cycle course/s as entry requirements

Learning outcomes

The overall goal of the course is for students to aquire a better understanding of Python programming and its more advanced concepts. Building on basic knowledge in Python the course addresses central aspects of code quality, object-oriented programming (OOP), and best practices for software development in scientific environments.

Knowledge and understanding

After completion of the course, the student shall be able to

- explain key elements of modern software engineering such as revision control systems and continuous integration

- account for fundamental concepts for solving linear and nonlinear systems of equations
- account for fundamental concepts of data fitting, including model selection, parameter estimation, and methods for evaluating fit quality
- describe the usage of AI in programming including its risks and benefits.

Competence and skills

After completion of the course, the student shall be able to

- write Python programs applying advanced programming concepts and data structures
- convert algorithms into efficient programming code
- use advanced plotting techniques, including 3D plotting, animations, as well as fitting and presentation of statistical data
- carry out a major programming project in groups and use modern code sharing practices
- present solutions to problems and numerical results in speech, writing, and graphic form
- document code following standard practice.

Judgement and approach

After completion of the course, the student shall be able to

- reflect and argue about the advantages and drawbacks of the Python programming language for high-performance computing
- critically analyse programs produced by fellow students and assess alternatives to their programming solution
- reflect on their own programming skills and ways to strengthen capabilities.

Course content

The course treats:

- Advanced programming concepts, data structures as well as code generation and multiprocessing
- Problem-solving using numerical methods associated with mathematics and physics
- Basic modules such as NumPy, SciPy and Matplotlib including different applications
- Data fitting with quality control and programming related to basic statistics
- Application of specialized modules such as Pandas (or equivalent), Numba (or equivalent), or pyCUDA (or equivalent) for efficient Python programming
- Advanced object oriented concepts (OOP), such as dataclasses and other concepts relevant for scientific applications
- Tests and profiling including continuous integration and revision control systems, primarily git

- Good practice of coding and documentation.

Course design

The teaching consists of lectures and seminars. Compulsory computer exercises as well as a larger programming project that is carried out in smaller student groups are included in the course.

Assessment

The examination consists of oral presentations of computer exercises during the course, a written report and oral presentation of a larger programming project completed in groups, as well as an individual written exam in the form of a computer-based test at the end of the course.

Students who do not pass the ordinary examination are offered a resit examination shortly thereafter.

The examiner, in consultation with Disability Support Services, may deviate from the regular form of examination in order to provide a permanently disabled student with a form of examination equivalent to that of a student without a disability.

Grades

Grading scale includes the grades: Fail, Pass
To obtain the grade pass on the entire course requires passing the computer exercises, the written and oral presentation of the programming project, as well as the computer-based test.

Entry requirements

Admission to the course requires 30 credits in Science, (at least reviewed, but not necessarily completed), including:

- basic knowledge in Mathematics as provided by MATA31 Analysis in One Variable, 15 credits, and MATA32 Algebra and Vector Geometry, 7.5 credits or equivalent,
- basic knowledge in Python as provided by MATA31 Analysis in One Variable, 15 credits or MNXA22 Computer Literacy, 3 credits, or equivalent,
- basic knowledge of data analysis as provided by FYSA33 Electricity and Magnetism, 7.5 credits or equivalent.

Further information

The course cannot be included in a degree together with the course NUMA01 Computational Programming with Python, 7.5 credits.

The course is given by the Centre for Mathematical Sciences, Lund University.