
CO1102 Programming Fundamentals

Credits: 15 **Convenor:** Dr E. Tadjouddine & Dr R. Zare **Semester:** 1st

Prerequisites: *Essential:* -
 Desirable: -

Lectures: 20 *hours*

Tutorials: 10 *hours*

Laboratories: 20 *hours*

Independent Study: 100 *hours*

Assessment: *Coursework: 100%*

Formative Coursework

None

Summative Coursework

Class Tests: 2 *in total*

Assignments: 1 *in total*

Learning Outcomes Students should be able to:

- Explain the fundamentals of imperative programming and write elementary programs.
 - Analyse simple problems and write solution programs using variables, types, expressions and basic operators, conditional and looping control structures, functions and I/O and exceptions.
 - Describe techniques for simple software design and development using very simple algorithms and data structures.
 - Write simple programs involving text and file I/O, and graphics interfaces, and data types such as strings, numbers, lists, tuples. Make use of editors and development environments.
 - Describe fundamentals of OO programming and write simple OO programs using classes and objects
-

Explanation of Prerequisites No specific knowledge is required, but a very rudimentary understanding of logic and discrete mathematics will be helpful.

Module Description Programming is a creative activity that can be fun and/or help us solve a wide range of problems including data analysis or artificial intelligence. We are increasingly surrounded by devices that need be programmed in some language. Python was conceived in the late 1980s as a scripting and teaching language but has gained in popularity for many programmers including engineers and data scientists across academia and industry.

In this module, we aim to provide fundamental skills in writing computer programs by using the latest version of Python. We focus on solving IT-related problems and introduce the building blocks of a computer program: inputs/outputs, sequential/conditional/repeated execution, and reuse. We will then introduce the concepts of classes and objects. Throughout, we emphasise on the importance of the syntax and semantics of the Python programming language as well as the design, implementation and testing of a computer program. The concepts introduced in this module will be revisited and further in the upcoming modules CO1105 and CO1107.

Syllabus

- Introduction: Basic Linux commands, editors, interpreter vs. compiler, and Python IDEs
- From an algorithm to a computer program
- Basic built-in types
- Variables, expressions, assignments and basic IO

- Logical expressions and conditional execution
- Testing and Debugging programs
- Basic Python packages and using functions
- Defining functions
- Iteration - loop patterns
- Python Turtle Graphics
- Basic built-in data types - Strings, Arrays, Lists, Sets and Dictionaries
- Introduction to files and exception handling
- Introduction to object-oriented programming

Reading List

- [B] Runestone Academy, *How to Think Like a Computer Scientist: Interactive Edition*, see <https://runestone.academy/runestone/default/user/login?next=/runestone/default/index> , .
- [B] Zed A. Shaw, *Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code*
ISBN-13: 978-0134692883 , Zed Shaw's Hard Way Series.
- [B] Dusty Phillips, *Python 3 Object-oriented Programming - Second Edition*
ISBN: 9781784398781, PACKT Publishing.
- [B] Albert Schueller, *How to Think Like a Computer Scientist: Learning with Python 3*, The Whitman Local Edition (WLE).

Convenor's Notes There are two class tests and one assignment.

CW1 Class Test, 20% module mark

CW2 Take Home Worksheet, 40% module mark

CW3 Class Test, 40% module mark