

University of Roehampton
International Summer School

School of Arts**Module Title: Software Development 1****Module Code: CMP020L001H****Module Rationale**

Software Development 1 introduces students to the fundamental concepts, methodologies, and techniques of software development. Programming is a key component of computer science and is an in-demand skill for the workplace inside and outside of the IT industry. Software Development 1 introduces the fundamental principles of software development, including syntax and semantics, variables and primitive data, expressions and assignment, input-output, conditions, iteration, functions, recursion, and an introduction to algorithms. The module details how to build programs using these techniques and how to apply problem-solving strategies in the design and implementation of simple programs. Students will practise the skills of programming. They will work in a high-level language, using the tools to design, implement, build, execute, and test software applications.

Software Development 1 provides students with core programming competencies. The aim of Software Development 1 is to develop students' fluency in programming languages and software development. The module will require students to both implement their own programs and trace the behaviour of existing programs.

Learning Outcomes

Students who successfully complete this module will:

- Design, implement, test, and debug a program that uses each of the following fundamental programming constructs – basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing.
- Analyse and explain the behaviour of simple programs involving the fundamental programming constructs – variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion.
- Identify the relative strengths and weaknesses among multiple designs or implementations for a problem.
- Use a programming language to implement, test, and debug algorithms for solving simple problems.

Topics:

- Introduction to programming.
- Basic syntax and semantics of a higher-level language.
- Variable and primitive data types (numbers, characters, Booleans).
- Expressions and assignments.
- Simple I/O including file I/O.
- Conditional and iterative control structures.
- Functions and parameter passing.

- Recursion.
- Introduction to algorithms.
- Problem solving with algorithms.
- Introduction to algorithmic problem-solving strategies.
- Fundamental design concepts and principles (abstraction, problem decomposition, encapsulation).

Assessment

Coursework (program) – 60%

Class test – 40%

Essential Reading

Python Programming: An Introduction to Computer Science, 3rd Ed. - John Zelle, Aug 08, 2016

The Python Workbook: A Brief Introduction with Exercises and Solutions - Ben Stephenson, 2016