# Introduction to Programming (Python)

# EE115

# 5 ECTS Credits

| Module Name | Introduction to Programming [ Python] |
|---|---|
| Module Code | EE115 |
| Module Co-ordinator | Refer to Excel document *Module_Co-ordinators* |
| Department | Electronic Engineering |
| Credit rating | 5 ECTS Credits |
| Pre-requisites | None |

| Aims | The main aim of this module is to introduce basic problem solving techniques, approaches to planning and organizing computer programs, and the common computer programming language elements used to express the task to be performed when implementing a computer program. |
|---|---|
| Learning Outcomes | At the end of the course, the student should be able to:<br><br>1. Design, write and explain simple computer programs.<br>2. Communicate algorithms effectively using comments and appropriate naming of identifiers.<br>3. Use literal values, constants, and variables.<br>4. Make use of Boolean expression appropriately.<br>5. Use selection and iteration appropriately.<br>6. Define and use functions appropriately.<br>7. Make use of libraries (Python "modules").<br>8. Use basic file input/output functions. |

**Time Allowance for Constituent Elements**

| | |
|---|---|
| Lectures | 21 hours |
| Assignments (10 x 6 hr, including 10 x 3hr lab sessions) | 60 hours |
| Laboratory exams (2 x 3 hr) | 6 hours |
| Tutorials (8 x 1 hr) | 8 hours |
| Independent study | 28 hours |

| Semester Examination | 2 hours |
|---|---|

**Indicative Syllabus**

- How to launch python and write python scripts
- Programming fundamentals: variables, types, expressions and assignment
- Creating and printing text to output window
- Creating variable names of various types and assigning values
- Reading in user data
- Converting between types
- Simple mathematical operations using integers and floats; programming with complex numbers
- Basic comparisons using different types
- Boolean operations
- Loops, "if" and "else" statements
- Formatting programs: indentation and comments
- Creating and using lists
- Making use of list indices, lengths, changing list items and extending lists
- Using list methods including creating and testing lists and removing from lists
- Using list iterators; Using lists inside loops; Making use of list "slices"
- Reading from and writing to files
- Writing and using functions
- Using "tuples"
- Understanding "modules" (libraries)
- Finding and making use of system modules and external modules
- Python "dictionaries"
- Gentle introduction to classes and inheritance

**Assessment Criteria**

| Semester Examination | 50% |
|---|---|
| Assignments (10) | 20% |
| Lab exams | 30% |

**Penalties:** Class and laboratory tests cannot, in general, be repeated.

**Pass Standard and any Special Requirements for Passing Modules:** In order to pass this module, students must achieve an overall mark of at least 40%. Students are not required to pass the written and continuous components separately.

**Supplemental Examination**: 1 x 2 hour written examination (Autumn). The continuous assessment mark is carried forward as there is no facility for repeating the continuous assessment elements of the course.

**Continual Assessment Results:** All coursework elements will be corrected within two weeks, where that does not extend past the end of the semester. Results and corrected scripts will be available for viewing upon request.

---

**Assessment Philosophy**

This course is aimed at those that are entirely new to programming and provides an introduction to programming using Python. By the end of this course, students should be able to write simple Python programs and to interpret more complex Python programs written by others. This course is intended as a gentle introduction to the basic fundamentals of programming with minimal emphasis on setting up the programming environment. The course is designed as a simple lead-in to 1$^{st}$ years EE108 that deals with embedded C and 2$^{nd}$ years EE208 module, which deals with object oriented programming using C++ and employs more rigorous debugging tools. The assignments, lab exams, and semester examination all cover all learning outcomes.

---

| | |
|---|---|
| **Course Text** | Pilgrim, Mark. Dive Into Python. Apress, 2004. (Available free online) |
| **References** | The official Python documentation can be found on-line at: http://docs.python.org/ |

---

| Programmes currently utilising module | Compulsory |
|---|---|
| BE in Electronic Engineering | Yes |