

This Version is No Longer Current
 The latest version of this module is available [here](#)

MODULE DESCRIPTOR

Module Title

Object Oriented Software Development

Reference	EN3543	Version	4
Created	July 2017	SCQF Level	SCQF 9
Approved	December 2009	SCQF Points	15
Amended	August 2017	ECTS Points	7.5

Aims of Module

To provide the student with the skills and knowledge necessary to manage software projects and to develop software systems using an object-oriented approach.

Learning Outcomes for Module

On completion of this module, students are expected to be able to:

- 1 Design, implement and test software using an object-oriented approach.
- 2 Utilise appropriate rapid application development tools in the development of a software solution.
- 3 Design, implement and test software using appropriate data structures.
- 4 Design and implement a graphical user interface.

Indicative Module Content

Project Administration: size, time and cost considerations of a software project. Project planning, life cycle. Project control: progress monitoring, revision control. Project maintenance. Object-oriented Design: objects, object classes, inheritance. Design representation, UML. Object-oriented Programming: classes, private, protected and public data and member functions. Constructors and destructors. Overloading, derived classes, virtual functions. Software Development for graphical user interfaces: Class frameworks, rapid application development tools, user interface design, widgets, event handling. Review of Programming Languages: review of the features of current programming languages which are applicable to electronic engineering applications. Case studies. Data Structures: linked lists, doubly-linked lists, queues, trees, stacks, standard template library.

Module Delivery

The module is taught using a structured programme of lectures, tutorials, laboratories and student-centred learning.

Indicative Student Workload	Full Time	Part Time
Contact Hours	36	N/A
Non-Contact Hours	114	N/A
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	150	N/A
<i>Actual Placement hours for professional, statutory or regulatory body</i>		

ASSESSMENT PLAN

If a major/minor model is used and box is ticked, % weightings below are indicative only.

Component 1

Type: Coursework Weighting: 30% Outcomes Assessed: 1
 Description: Coursework involving the development and testing of software using C++.

Component 2

Type: Coursework Weighting: 70% Outcomes Assessed: 2, 3, 4
 Description: Coursework involving the development of and testing of software, based around a GUI, using complex data structures.

MODULE PERFORMANCE DESCRIPTOR

Explanatory Text

To pass the module, you must achieve a 40% weighted average mark from the two courseworks. In addition, you need to achieve at least 35% in each coursework Component.

Module Grade	Minimum Requirements to achieve Module Grade:
A	70-100%
B	60-69%
C	50-59%
D	40-49%
E	35-39%
F	0-34%
NS	Non-submission of work by published deadline or non-attendance for examination

Module Requirements

Prerequisites for Module	Computer Architecture (EN3540) or equivalent.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

- 1 SUMMERFIELD, M., 2010. Programming in Python 3. 2nd ed. Addison Wesley.
- 2 KALB, I., 2018, Learn to Program with Python 3, Apress
- 3 SUNDRES, J., 2020, Introduction to Scientific Programming with Python, Springer Open
- 4 NAGAR, S., 2017, Introduction to Python for Engineers and Scientists, Apress
- 5 DONALDSON, T., 2013, Python: Visual QuickStart Guide, 3rd Ed, Peachpit Press
- 6 SHOVIC, J.C., 2021, Python All-In-One For Dummies, 2nd ed. For Dummies