

MODULE DESCRIPTOR

Module Title

Object-oriented Programming

Reference	CMM024	Version	4
Created	February 2023	SCQF Level	SCQF 11
Approved	July 2016	SCQF Points	15
Amended	June 2023	ECTS Points	7.5

Aims of Module

To master fundamental concepts and practical skills in procedural programming. To explore the key concepts of object-oriented programming. To develop the student's skill in the practical design, development, and testing of object-oriented applications. To extend the student's knowledge and proficiency in object-oriented design to include class design by inheritance and to understand a given design presented in a notation.

Learning Outcomes for Module

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

- 1 Produce a software solution to a problem using key constructs and mechanisms in a procedural programming language.
- 2 Analyse a set of requirements to create an object-oriented design.
- 3 Create a working solution from a design using an object-oriented programming language.
- 4 Effectively communicate an object-oriented design using an appropriate notation.

Indicative Module Content

The module introduces basic procedural programming concepts and mechanisms, including variables, decision control, repetition, arrays, etc. The module will then explore essential object-oriented concepts and techniques in the design of class hierarchies. Key concepts of object-oriented programming include classes, objects, instance variables, methods, and the use of hierarchical mechanisms for object re-use through composition, association, inheritance, interfaces, abstract classes, and polymorphism. An appropriate notation will be introduced to guide and document the design.

Module Delivery

Key concepts are introduced and illustrated in lectures. The laboratory sessions are used to develop and evaluate design ideas before implementation and then to progress through a series of exercises intended to develop understanding of the lecture content and to develop proficiency in the practical application of object-oriented programming skills.

Indicative Student Workload

	Full Time	Part Time
Contact Hours	30	30
Non-Contact Hours	120	120
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	150	150
<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:	Practical Exam	Weighting:	100%	Outcomes Assessed:	1, 2, 3, 4
Description:	Computer-based programming assessment.				

MODULE PERFORMANCE DESCRIPTOR**Explanatory Text**

The calculation of the overall grade for this module is based on combined marks from all parts from a multi part practical programming assessment.

Module Grade	Minimum Requirements to achieve Module Grade:
A	A grade in Component 1
B	B grade in Component 1
C	C grade in Component 1
D	D grade in Component 1
E	E grade in Component 1
F	F grade in Component 1
NS	Non-submission of work by published deadline or non-attendance for examination

Module Requirements

Prerequisites for Module	None.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

- 1 DEITEL, P.J. AND DEITEL, H., 2015. Java: Early Objects 10th ed. Prentice Hall.
- 2 SCHILDT, H., 2019. Java: The Complete Reference. 11th ed. McGraw Hill.
- 3 LIANG, Y, D., 2019. Introduction to Java Programming and Data Structures. Comprehensive version. Pearson.
- 4 HORSTMANN, C., 2016. Core Java. 10th ed. Prentice Hall.
- 5 HORSTMANN, C, 2016. Big Java: Late Objects. 2nd ed. Wiley.