MODULE DESCRIPTOR

Module Title

Object Oriented Programming

Reference	CMM504	Version	5
Created	April 2017	SCQF Level	SCQF 11
Approved	April 2005	SCQF Points	15
Amended	August 2017	ECTS Points	7.5

Aims of Module

To extend the student's knowledge and proficiency in object oriented design to include class design by inheritance. To introduce issues of object oriented design that arise in the develoment of interactive programs that incorporate a graphicial user interface.

Learning Outcomes for Module

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

1	Demonstrate extended knowledge and understanding of object oriented design concepts, inheritance, interface and abstract classes.
2	Apply the principle of class inheritance (in addition to composition and association) to construct hierarchies of new classes including components required for graphical interfaces.
3	Use an event handling model to identify components and interactions required to design an interactive object oriented program.
4	Implement object oriented programs that incorporate a graphical user interface.

Indicative Module Content

A major theme of the module will be the use of constructive object oriented techniques (especially inheritance) in the design of class hierarchies. The module will focus on the particular application of inheritance to the design and implementation of interactive object oriented programs that incorporate a graphical user interface. Module content will cover inheritance, interfaces, abstract classes, polymorphism, graphical toolkits, event handling model, graphical interfaces for applications and applets, exception handling. The module content will continue to emphasise use of an appropriate methodolgy (UML) to guide and document the design process.

Module Ref:	CMM504 v5

Module Delivery

Key concepts and ideas are introduced in lectures. Tutorials are used to develop and evaluate design ideas (using Unified Modelling Language) before implementation. In the lab sessions, the students will learn practical aspects of object oriented programming including the use of existing packages for development of graphical user interfaces and programming tools that aid the development process.

Indicative Student Workload	Full Time	Part Time
Contact Hours	44	44
Non-Contact Hours	106	106
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	150	150
Actual Placement hours for professional, statutory or regulatory body		

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:	A multi-part practical exam				

MODULE PERFORMANCE DESCRIPTOR

Explanatory Text

Student must achieve a grade D or better to pass the module. Marks from parts of the practical exams are combined to calculate the module grade.

Module Grade	Minimum Requirements to achieve Module Grade:
A	The student needs to achieve an A in C1.
B	The student needs to achieve an B in C1.
C	The student needs to achieve an C in C1.
D	The student needs to achieve an D in C1.
E	The student needs to achieve an E in C1.
F	The student needs to achieve an F in C1.
NS	Non-submission of work by published deadline or non-attendance for examination

Module Requirements

Prerequisites for Module	None in addition to course requirements.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

1	DEITEL, P. AND DEITEL, J., 2015. Java: How to program. Pearson Education.
2	CORNELL, G. AND HORSTMANN, C., 2008. Core Java 2 - Volume 1 Fundamentals. Prentice Hall.
3	STEVENS, P. AND POOLEY, R., 2000. USING UML: Software Engineering with Objects and Components. Addison Wesley.