MODULE DESCRIPTOR

Module Title

Object Oriented Software Design

Reference	CM2015	Version	5
Created	October 2017	SCQF Level	SCQF 8
Approved	July 2008	SCQF Points	30
Amended	October 2017	ECTS Points	15

Aims of Module

To extend the student's knowledge and proficiency in object oriented design, and to provide the student with the ability to apply concepts of algorithm and data structure design, analysis and implementation.

Learning Outcomes for Module

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

1	Demonstrate an extended knowledge and understanding of object oriented design concepts concerning inheritance, interfaces, 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 interaction required to design and implement object oriented programs that incorporate a graphical user interface.
4	Analyse and make a critical comparison between alternative designs of algorithms and data structures.
5	Design appropriate and efficient implementations for a number of commonly occurring data abstractions.

Indicative Module Content

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, in this area, will cover: inheritance, interfaces, abstract classes, polymorphism, graphical toolkits, event handling model, graphical interfaces for applications and applets, exception handling, design patterns, testing. The second half of the Module will focus on algorithms and data structures. Module content will include: worst/average/best case characteristics of algorithms. Implementation of standard data abstractions using: arrays, lists, trees, hash tables. Strategies for algorithm design. Collection frameworks.

Module Ref:	CM2015 v5

Module Delivery

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

Indicative Student Workload	Full Time	Part Time
Contact Hours	126	N/A
Non-Contact Hours	174	N/A
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	300	N/A
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:	Coursework	Weighting:	100%	Outcomes Assessed:	1, 2, 3, 4, 5
Description:	A practical piece of coursework.				

MODULE PERFORMANCE DESCRIPTOR

Explanatory Text

The calculation of the overall grade for this module is based on 100% weighting for C1. An overall minimum grade D is required to pass the module.

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

Module Requirements

Prerequisites for Module	The student will normally be expected to have completed the module CM1015 Software Design and Development or equivalent.
Corequisites for module	None.
Precluded Modules	None.

Module Ref:	CM2015 v5

INDICATIVE BIBLIOGRAPHY

1	DEITEL,P. and DEITEL, H.,2014. Java:How to Program (Late objects). 10th ed. Prentice Hall.
2	LIANG, Y. D, 2013. Introduction to Java Programming.9th ed. Pearson
3	HORSTMANN, C, 2013. Big Java: Late Objects. 1st ed. John Wiley.
4	SAVITCH, W., 2013. Absolute Java. 5th ed. Pearson.
5	GOODRICH and TAMASSIA. 2005. Data Structures and Algorithms in Java. 4th ed. John Wiley.