

<b>Module Title</b> <b>Object Oriented Software Design</b>	Reference CM2015 SCQF Level SCQF 8 SCQF Points 30 ECTS Points 15 Created May 2008 Approved July 2008 Amended September 2012 Version No. 2
<b>Keywords</b> Object Oriented Design, Object Oriented Programming Inheritance, Graphical User Interfaces, Algorithm Design, Lists, Trees, Hashed Tables	

## This Version is No Longer Current

The latest version of this module is available [here](#)

### Prerequisites for Module

The student will normally be expected to have completed the module CM1015 Software Design and Development or equivalent.

### Indicative Student Workload

<i>Contact Hours</i>	Full Time
Assessment	22
Laboratories	48
Lectures	44
Tutorials	12

### Corequisite Modules

None.

<i>Directed Study</i>	
Directed Study	74

### Precluded Modules

None.

<i>Private Study</i>	
Private Study	100

### Mode of Delivery

### 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.

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 fro

## 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

development of graphical user interfaces and programming tools that aid the development process.

## Assessment Plan

	Learning Outcomes Assessed
Component 1	1,2,3,4,5

Component 1 - Coursework

## 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.

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, reusable software component model (beans). The module content will continue to emphasise use of an appropriate methodology (UML) to guide and document the design process. 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.