

This Version is No Longer Current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

MODULE DESCRIPTOR

Module Title

Advanced Software Design and Development

Reference	CM2100	Version	2
Created	April 2017	SCQF Level	SCQF 8
Approved	July 2016	SCQF Points	30
Amended	August 2017	ECTS Points	15

Aims of Module

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

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 modelling and design concepts concerning inheritance, interfaces, and abstract classes.
- 2 Design appropriate and efficient implementations for commonly occurring data abstractions.
- 3 Identify and implement simple examples of selected software system classes.
- 4 Use an event handling model to identify components and interaction required to design and implement object oriented programs that incorporate a graphical user interface.
- 5 Analyse, and make a critical comparison between, alternative designs of algorithms and data structures based on efficiency, scalability and security concerns.

Indicative Module Content

Inheritance, interfaces, abstract classes, polymorphism, graphical toolkits, event handling model, graphical user interfaces, exceptions, file handling, testing. Design Patterns Algorithms. Sorting and searching, recursive algorithms, computational complexity. Implementation of standard data abstractions using: arrays, lists, trees, hash tables. Strategies for algorithm design. Collection frameworks. Security aspects of advanced software development processes. Standards and Best Practice Guides: ISO 27001, ISO 27014, BSSIM SSF.

Module Delivery

The module will be delivered through a mixture of lectures, tutorials and laboratory sessions.

Indicative Student Workload

	Full Time	Part Time
Contact Hours	96	N/A
Non-Contact Hours	204	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:	Practical Exam	Weighting:	60%	Outcomes Assessed:	1, 2, 3
Description:	Practical time-limited coding assessment.				

Component 2

Type:	Coursework	Weighting:	40%	Outcomes Assessed:	4, 5
Description:	Extended software development exercise.				

MODULE PERFORMANCE DESCRIPTOR**Explanatory Text**

The calculation of the overall grade for this module is based on a 60% weighting for C1 and 40% weighting for C2. An overall minimum grade D is required to pass the module.

		Practical Exam:						
		A	B	C	D	E	F	NS
Coursework:	A	A	B	B	C	C	E	
	B	A	B	C	C	D	E	
	C	B	B	C	D	D	E	
	D	B	C	C	D	E	E	
	E	C	C	D	D	E	F	
	F	E	E	E	E	E	F	
	NS	Non-submission of work by published deadline or non-attendance for examination						

Module Requirements

Prerequisites for Module	CM1104 Object Oriented Programming, or equivalent.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

- 1 Horstmann, C., 2016. Big Java: Late Objects. 2nd Ed. John Wiley
- 2 Savitch, W., 2016. Absolute Java. 6th Ed. Pearson
- 3 GOODRICH, M.T. and TAMASSIA, R., 2014. Data Structures and Algorithms in Java. 6th Ed. John Wiley.
- 4 Liang, Y.D., 2017. Introduction to Java Programming and Data Structures. Comprehensive Version. 11th ed. John Wiley
- 5 Deitel, P. and Deitel, H., 2017. Java: How to Program (Late Objects). 11th Ed. Prentice Hall
- 6 SOMMERVILLE, I., 2015. Software Engineering. 10th Ed. Pearson.