ROBERT GORDON UNIVERSITY ABERDEEN

This Version is No Longer Current
The latest version of this module is available here

MODULE DESCRIPTOR

Module Title

Object Oriented Programming

Reference	CM1104	Version	2
Created	April 2017	SCQF Level	SCQF 7
Approved	July 2016	SCQF Points	15
Amended	August 2017	ECTS Points	7.5

Aims of Module

None.

Learning Outcomes for Module

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

1	Identify the main elements of an object oriented language and describe the software development problems that these features are meant to overcome.
2	Use an API to design, implement and test solutions to software development problems using interacting objects.
3	Make use of standard collection structures, together with appropriate algorithms, to solve programming problems.
4	Use an appropriate modelling language to develop diagrammatic representations of problems in order to illustrate relations and interactions between software objects.
5	Identify and assess security concerns and mitigate elements of risk in the software development process.

Indicative Module Content

Software Basics: Variable, data types, declarations and expressions, iterative and conditional programming constructs, methods, recursion. OO Concepts: Encapsulation, abstraction, data hiding, inheritance, polymorphism, code reuse. Security aspects of object oriented software development. OO Programming: Classes and objects, arrays, exception handling, defensive programming, simple data structures, Application Programming Interfaces. Modelling and Design: UML class diagrams, interaction diagrams. Standards and Best Practice Guides: ISO 27001, ISO 27014.

Module Delivery

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

Indicative Student Workload

	Full Time	Part Time
Contact Hours	60	N/A
Non-Contact Hours	90	N/A
Placement/Work-Based Learning Experience [Notional] Hours	N/A	N/A
TOTAL	150	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:	100%	Outcomes Assessed:	1, 2, 3, 4, 5

Description: The module will be assessed by practical examination of the key skills and competencies.

MODULE PERFORMANCE DESCRIPTOR

Explanatory Text

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

Module Grade	Minimum Requirements to achieve Module Grade:
A	The student needs to achieve an A in C1.
B	The student needs to achieve a B in C1.
C	The student needs to achieve a C in C1.
D	The student needs to achieve a 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	CM1100 Introduction to Computing, or equivalent.
Corequisites for module	None.
Precluded Modules	None.

INDICATIVE BIBLIOGRAPHY

1	HORSTMANN, C., 2013. Big Java: Late Objects. 1st Ed. John Wiley.
2	SAVITCH, W., 2013. Absolute Java. 5th Ed. Pearson.
3	LIANG, Y. D., 2013. Introduction to Java Programming. 9th Ed. Pearson
4	DEITEL,P. and DEITEL, H., 2012. Java: How to Program. 9th Ed. Prentice Hall.
5	SOMMERVILLE, I., 2015. Software Engineering. 10th Ed. Pearson.
6	MCLAUGHLIN, B., POLLICE, G. and WEST,D., 2009. Head First Object-Oriented Analysis and Design. Safari Books Online, 978-0-596-55675-4