



Title:	Object Oriented Programming	APPROVED
Long Title:	Object Oriented Programming	
Module Code:	COMP7013	Duration: 1 Semester
Credits:	5	
NFQ Level:	Intermediate	
Field of Study:	Computer Science	
Valid From:	Semester 1 - 2017/18 (September 2017)	
Module Delivered in	4 programme(s)	
Module Coordinator:	Sean McSweeney	
Module Author:	Paul Davern	
Module Description:	Students will gain extensive experience of using an Object Oriented Programming Language by building an application from a case study/well-defined problem. The focus of the module is to give the student practical experience of using a number of class libraries. The student will gain an understanding of the runtime abstractions that are associated with an Object Oriented programming language.	
Learning Outcomes		
<i>On successful completion of this module the learner will be able to:</i>		
LO1	Explain the runtime organisation of an Object Oriented software program.	
LO2	Use a debugger and testing frameworks in the development of a software program.	
LO3	Apply design patterns and organize code using best practices in the development of a software program developed using an Object Oriented programming language.	
LO4	Use class libraries that are packaged with an Object Oriented programming language, such as collection classes, GUI classes, and database access classes.	
LO5	Develop a substantial software application using class libraries and design patterns.	
Pre-requisite learning		
Module Recommendations		
<i>This is prior learning (or a practical skill) that is strongly recommended before enrolment in this module. You may enrol in this module if you have not acquired the recommended learning but you will have considerable difficulty in passing (i.e. achieving the learning outcomes of) the module. While the prior learning is expressed as named CIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).</i>		
12784	SOFT7004	Object Oriented Principles
12793	COMP7013	Object Oriented Programming
Incompatible Modules		
<i>These are modules which have learning outcomes that are too similar to the learning outcomes of this module. You may not earn additional credit for the same learning and therefore you may not enrol in this module if you have successfully completed any modules in the incompatible list.</i>		
No incompatible modules listed		
Co-requisite Modules		
No Co-requisite modules listed		
Requirements		
<i>This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed. You may not enrol on this module if you have not acquired the learning specified in this section.</i>		
No requirements listed		



Module Content & Assessment

Indicative Content

Runtime environment and Operating System Interaction

Object Life Cycle: class loader, garbage collection; Destroying and Finalizing Objects; Object Identity vs. Equality, Aliasing; Class Objects; Introduction to reflection; Operating System interaction: file system manipulation, system calls, stream processing;

Debugging and Testing

Using a debugger: stack and heap, stack variables, globals; Memory leaks; Testing Frameworks.

Patterns and Code Organisation

Packages; Build Systems; Interfaces; Creational Patterns: Singleton, Abstract factory; Observer - event-driven programming; Model-View Controller; Separation of concerns; Weak coupling.

The Design of Class Libraries

Understanding the design of the standard Class Libraries that are packaged with an Object Oriented programming language: using the API, hierarchy of classes, navigating source code and documentation; Generics; Building and distributing libraries; Using non-standard class libraries.

Using Class Libraries - Collection, GUI and database access

Collection class library: Iterator, HashMap, Set, Linked Lists; Trees; GUI class library: GUI Components, Event driven programming, introduction to threads, Asynchronous and synchronous programming; SQL client class library: database queries, database cursors.

Assessment Breakdown

%

Course Work

100.00%

Course Work

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Practical/Skills Evaluation	Practicals based on lectures. For example: a) the development of application which generates a cross reference listing for a inputted source code file. b) the development of a hierarchical data structure such as a family tree.	1,2,3,4	30.0	Every Second Week
Short Answer Questions	This assessment will examine students on material presented in class.	1,3,4	30.0	Week 7
Project	The student will be expected to develop a software application. For example, develop an application which uses class libraries such as containers, GUI and SQL client.	1,2,3,4,5	40.0	Sem End

No End of Module Formal Examination

Reassessment Requirement

Coursework Only

This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.

The institute reserves the right to alter the nature and timings of assessment

Module Workload

Workload: Full Time				
<i>Workload Type</i>	<i>Workload Description</i>	<i>Hours</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Presentation of indicative content.	2.0	Every Week	2.00
Lab	Practical work based on indicative content.	2.0	Every Week	2.00
Independent & Directed Learning (Non-contact)	Self Study.	3.0	Every Week	3.00
Total Hours				7.00
Total Weekly Learner Workload				7.00
Total Weekly Contact Hours				4.00

Workload: Part Time				
<i>Workload Type</i>	<i>Workload Description</i>	<i>Hours</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Presentation of indicative content.	2.0	Every Week	2.00
Lab	Practical work based on indicative content.	2.0	Every Week	2.00
Independent & Directed Learning (Non-contact)	Self Study.	3.0	Every Week	3.00
Total Hours				7.00
Total Weekly Learner Workload				7.00
Total Weekly Contact Hours				4.00

Module Resources
<i>Recommended Book Resources</i>
<ul style="list-style-type: none"> • Harvey, Deitel, Deitel 2014, <i>Java, How to Program</i>, 10th Ed., Pearson [ISBN: 9780133807806]
<i>Supplementary Book Resources</i>
<ul style="list-style-type: none"> • Benjamin J Evans, David Flanagan 2014, <i>Java in a Nutshell</i>, 6th Ed., O'Reilly Media [ISBN: 9781449370824] • Lewis and Loftus 2014, <i>Java Software Solutions</i>, 8th Ed., Pearson [ISBN: 9781292018232]
<i>This module does not have any article/paper resources</i>
<i>Other Resources</i>
<ul style="list-style-type: none"> • Website: <i>OracleJava Tutorial from Oracle</i> http://docs.oracle.com/javase/tutorial/ • Website: <i>Design Patterns</i> http://www.oodesign.com/ • Website: <i>Standard Java Class Library</i> https://docs.oracle.com/javase/7/docs/api/ • Website: <i>java FX</i> http://docs.oracle.com/javafx/ • Website: <i>Collections documentation and tutorial</i> http://docs.oracle.com/javase/tutorial/collections/ • Website: <i>JDBC Database Access</i> http://docs.oracle.com/javase/tutorial/jdbc/index.html

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	<u>Bachelor of Science (Honours) in Software Development</u>	4	Mandatory
CR_KDNET_8	<u>Bachelor of Science (Honours) in Computer Systems</u>	4	Mandatory
CR_KCOMP_7	<u>Bachelor of Science in Software Development</u>	4	Mandatory
CR_KCOME_6	<u>Higher Certificate in Science in Software Development</u>	4	Mandatory