



Title:	Object Oriented Principles APPROVED
Long Title:	Object Oriented Principles
Module Code:	SOFT7004
Duration:	1 Semester
Credits:	5
NFQ Level:	Intermediate
Field of Study:	Computer Software
Valid From:	Semester 1 - 2017/18 (September 2017)
Module Delivered in	6 programme(s)
Module Coordinator:	Sean McSweeney
Module Author:	DENIS LONG
Module Description:	Object Oriented Programming is a programming paradigm where data and operations can be defined together as objects and allows reuse of these objects within the same or other programs. In this module students will learn object creation and design; how to store objects in an array or data structures; exceptions; basic file input, output and testing.
Learning Outcomes	
<i>On successful completion of this module the learner will be able to:</i>	
LO1	Discuss the fundamental principles of Object Oriented Programming (OOP).
LO2	Identify the benefits of object-oriented design and implementation.
LO3	Design and develop a program from a high level specification.
LO4	Use exception handling to deal with unexpected events that occur during the execution of a program.
LO5	Using the appropriate library in the Object Oriented Programming language, persist and read data to and from a file.
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
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

Object-Oriented Principles

Basic OO principles; Anatomy of a class; Encapsulation(data hiding); Creating objects from classes; Overloading principle; Object interaction; Grouping objects; Wrapper classes for primitive types; Abstract classes - Inheritance (super/subclasses); Class cohesion and coupling; Basic polymorphism - superclass reference types; Overriding principle; Aggregation and composition. Associations: uni-directional and bi-directional. Benefits of the object-oriented development approach. Responsibility driven design.

Testing and Debugging

Black Box, White Box & stress testing; Unit testing; Integration testing; Manual walk throughs.

Using Libraries

How to use the API and packages, including for example the String, Scanner, Random, Math, IO and the basic Collection class ArrayList.

Exceptions Handling

Definition of exceptions; The Exception class hierarchy; Unhandled exceptions; Try/catch mechanism; Throwing exceptions; Finally clause; The call stack; Exception handling needed to save data to a file.

File input and output

Writing basic text information to a file. Reading text from a file.

Practical Labs

The student will be required to develop and test object oriented programs based on the above content. Aspects of the above content will be demonstrated during these practical sessions.

Assessment Breakdown	%
Course Work	100.00%

Course Work				
<i>Assessment Type</i>	<i>Assessment Description</i>	<i>Outcome addressed</i>	<i>% of total</i>	<i>Assessment Date</i>
Multiple Choice Questions	Questions will be material presented in class.	1,2	20.0	Week 8
Practical/Skills Evaluation	The purpose of this assessment is to assess the lab work of the students. For example, the student may be required to develop a class or collection of classes that achieve a certain task, showing good object oriented principles throughout.	1,2,3,4	30.0	Every Second Week
Project	An example project may be to develop a class or collection of classes to achieve some requested functionality. The student may be expected to design an application composed of classes demonstrating some key principles and technologies of OOP, handle Exceptions, read and write to a file.	1,3,4,5	50.0	Week 13

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	Lecture delivering theory underpinning learning outcomes.	2.0	Every Week	2.00
Lab	Lab to support learning outcomes.	2.0	Every Week	2.00
Independent & Directed Learning (Non-contact)	Independent 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	Lecture delivering theory underpinning learning outcomes.	2.0	Every Week	2.00
Lab	Lab to support learning outcomes.	2.0	Every Week	2.00
Independent & Directed Learning (Non-contact)	Independent 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"> • Kathy Sierra, Bert Bates 2005, <i>Head first Java</i>, 2 Ed., O'Reilly [ISBN: 9780596009205]
<i>Supplementary Book Resources</i>
<ul style="list-style-type: none"> • Paul Deitel, Harvey Deitel 2014, <i>Java How to Program</i>, 10 Ed., Pearson [ISBN: 9780132575652] • Stuart Reges, Marty Stepp 2016, <i>Building Java programs: A Back to Basics Approach</i>, 4 Ed., Pearson Boston [ISBN: 9780134322766]
<i>This module does not have any article/paper resources</i>
<i>Other Resources</i>
<ul style="list-style-type: none"> • WebSite: ThenewCircle 2012, <i>Good learning resource</i>, TheNewCircle, Internet https://newcircle.com/bookshelf/java_fundamentals_tutorial/index • Website: Tutorialpoint 2017, <i>Good learning resource</i>, Tutorialpoint, Internet https://www.tutorialspoint.com/java/index.htm • WebSite: Bucky 2017, <i>Good Tutorial</i>, Bucky, US https://thenewboston.com/videos.php?cat=31 • WebSite: Oracle 2017, <i>Great tutorials</i>, Oracle, Internet http://docs.oracle.com/javase/tutorial/ • WebSite: Bluej 2017, <i>Development environment</i>, Bluej, Internet http://www.bluej.org/ • WebSite: Eclipse 2017, <i>Eclipse</i>, Eclipse, Internet http://www.eclipse.org/ • WebSite: Oracle 2017, <i>Development environment</i>, Oracle, Internet https://netbeans.org/kb/kb.html • WebSite: Java2s 2107, <i>Great Examples</i>, java2s, Internet http://www.java2s.com/ • Website: Derek Banas 2017, <i>Java Videos</i>, youtube, Youtube https://www.youtube.com/playlist?list=PLE7E8B7F4856C9B19

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	<u>Bachelor of Science (Honours) in Software Development</u>	3	Mandatory
CR_KDNET_8	<u>Bachelor of Science (Honours) in Computer Systems</u>	3	Mandatory
CR_KCOMP_7	<u>Bachelor of Science in Software Development</u>	3	Mandatory
CR_KCOME_6	<u>Higher Certificate in Science in Software Development</u>	3	Mandatory
CR_KCMSD_8	<u>Higher Diploma in Science in Cloud & Mobile Software Development</u>	1	Mandatory
CR_KCLCO_8	<u>Higher Diploma in Science in Cloud Computing</u>	1	Mandatory