

PROG C7005: Object-Oriented Programming

Module Details	
Module Code:	PROG C7005
Full Title:	Object-Oriented Programming APPROVED
Valid From:	Semester 1 - 2019/20 (June 2019)
Language of Instruction:	English
Duration:	2 Semesters
Credits:	10
Module Owner::	Bernadette Brosnan
Departments:	Unknown
Module Description:	Students completing this module will have a working knowledge of the principles and techniques involved in Object-Oriented Programming and will have developed their programming and problem solving skills.

Module Learning Outcome	
On successful completion of this module the learner will be able to:	
#	Module Learning Outcome Description
MLO1	Construct classes and methods.
MLO2	Implement a solution from a simple UML specification diagram.
MLO3	Discuss the criteria necessary for good OO program design.
MLO4	Develop and debug OO problem solutions.
MLO5	Apply appropriate object-oriented practices to problem solving.
MLO6	Use appropriate exception handling.
MLO7	Implement simple threading with an understanding of the impact of shared memory on programs.
Pre-requisite learning	
<p>Module Recommendations 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 DKIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).</p>	
No recommendations listed	

Module Indicative Content
Introduction Overview of object oriented principles, comparison with other paradigms.
User defined Classes Member data, modifiers, constructors, methods (accessor and mutator, general, overloaded), design -(cohesion, coupling and encapsulation).
Inheritance Basic inheritance, method overriding, polymorphism.
Collections Methods and application of classes and interfaces in collections framework.
Abstract Classes Abstract class definition, abstract methods.
Interfaces Implementation of library interfaces (e.g. Comparable & Comparator), writing own interfaces.
Exceptions Handling & propagating exceptions, throwing exceptions, writing own exceptions.
Threading Coding threads, shared memory, synchronization.
Generics Generic classes and type parameters, implementing generic types, generic methods.

Module Assessment

Assessment Breakdown	%
Course Work	60.00%
Final Examination	40.00%

Module Special Regulation

Assessments

Full Time

Course Work			
Assessment Type	Continuous Assessment	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 5	Learning Outcome	1,2,4
Duration in minutes	0		
Assessment Description Lab exam covering basic classes and inheritance			
Assessment Type	Continuous Assessment	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 12	Learning Outcome	1,2,4
Duration in minutes	0		
Assessment Description Lab exam covering inheritance, interfaces and abstract classes.			
Assessment Type	Class Test	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 24	Learning Outcome	1,6
Duration in minutes	0		
Assessment Description Lab exam			
Assessment Type	Class Test	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 28	Learning Outcome	3,5,6,7
Duration in minutes	0		
Assessment Description Multiple choice exam covering object oriented concepts and principles.			
Assessment Type	Continuous Assessment	% of Total Mark	20
Marks Out Of	0	Pass Mark	0
Timing	Week 29	Learning Outcome	1,4,5,6
Duration in minutes	0		
Assessment Description Staged pair project where students apply the concepts and techniques covered in class.			
No Project			
No Practical			
Final Examination			
Assessment Type	Formal Exam	% of Total Mark	40
Marks Out Of	0	Pass Mark	0
Timing	End-of-Semester	Learning Outcome	3,5,6,7
Duration in minutes	0		
Assessment Description Final exam covering all topics covered on the course. Focus on depth of understanding of theory.			
Part Time			

Course Work			
Assessment Type	Class Test	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 5	Learning Outcome	1,2,4
Duration in minutes	0		
Assessment Description Lab exam covering basic classes and inheritance			
Assessment Type	Class Test	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 12	Learning Outcome	1,2,4
Duration in minutes	0		
Assessment Description Lab exam covering inheritance, interfaces and abstract classes.			
Assessment Type	Class Test	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 24	Learning Outcome	1,6
Duration in minutes	0		
Assessment Description Lab exam			
Assessment Type	Class Test	% of Total Mark	10
Marks Out Of	0	Pass Mark	0
Timing	Week 28	Learning Outcome	3,5,6,7
Duration in minutes	0		
Assessment Description Multiple choice exam covering object oriented concepts and principles.			
Assessment Type	Continuous Assessment	% of Total Mark	20
Marks Out Of	0	Pass Mark	0
Timing	Week 29	Learning Outcome	1,4,5,6
Duration in minutes	0		
Assessment Description Staged pair project where students apply the concepts and techniques covered in class.			
No Project			
No Practical			
Final Examination			
Assessment Type	Formal Exam	% of Total Mark	40
Marks Out Of	0	Pass Mark	0
Timing	End-of-Semester	Learning Outcome	3,5,6,7
Duration in minutes	0		
Assessment Description Final exam covering all topics covered on the course. Focus on depth of understanding of theory.			

Module Workload

Workload: Full Time					
<i>Workload Type</i>	<i>Contact Type</i>	<i>Workload Description</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>	<i>Hours</i>
Lecturer-Supervised Learning (Contact)	Contact	There will be two 2-hour sessions of lab-based classes per week. In these sessions, theory and practical content will be integrated.	Every Week	4.00	4
Directed Reading	Non Contact	Students will be expected to spend 1 hour each week reading lecturer-recommended information sources, such as the recommended texts associated with the course.	Every Week	1.00	1
Independent Study	Non Contact	Students will be expected to undertake 3 hours of practical work (programming practice) each week, strengthening their programming abilities.	Every Week	3.00	3
Total Weekly Learner Workload					8.00
Total Weekly Contact Hours					4.00

Workload: Part Time					
<i>Workload Type</i>	<i>Contact Type</i>	<i>Workload Description</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>	<i>Hours</i>
Lecturer-Supervised Learning (Contact)	Contact	There will be two 2-hour sessions of lab-based classes per week. In these sessions, theory and practical content will be integrated.	Every Week	4.00	4
Directed Reading	Non Contact	Students will be expected to spend 1 hour each week reading lecturer-recommended information sources, such as the recommended texts associated with the course.	Every Week	1.00	1
Independent Study	Non Contact	Students will be expected to undertake 3 hours of practical work (programming practice) each week, strengthening their programming abilities.	Every Week	3.00	3
Total Weekly Learner Workload					8.00
Total Weekly Contact Hours					4.00

Module Resources

Supplementary Book Resources

Joel Murach. (2017), *Murach's Java Programming, 5*. Mike Murach & Associates Inc., [ISBN: 978-1-943872-].

Paul Dietel, Harvey Dietel. (2018), *Java How to Program, 11*. Pearson, [ISBN: 0-13-474335-0].

This module does not have any article/paper resources

Other Resources

[Web], Java API,
<https://docs.oracle.com>