APPROVED

COMP C8058: Programming 2

Module Details

Module Code:	COMP C8058
Full Title:	Programming 2 APPROVED
Valid From::	Semester 1 - 2019/20 (June 2019)
Language of Instruction:	English

Duration:	1 Semester

Credits::	10

Module Owner::	Tony McCarron

Departments:	Unknown

Module Description:	This module aims to expand students' understanding of basic OO programming techniques and provide them with a conceptual knowledge of the principles and techniques for software design.

Module Learning Outcome	
On successful completion of this module the learner will be able to:	
#	**Module Learning Outcome Description**
MLO1	Apply object-oriented programming in the use of external apis.
MLO2	Design, implement, unit-test and debug simple software systems.
MLO3	Document and implement solutions using simple UML specification diagrams
MLO4	Evaluate the suitability of adopting a particular design pattern to provide a solution for a given software development problem.
MLO5	Evaluate the need for refactoring when presented with existing solutions.

Pre-requisite learning
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).*
No recommendations listed

Module Indicative Content
-Class Inheritance Method overriding, abstract classes.
Implementation Implementation of existing interfaces, creation of new interfaces, rationale for interfaces over inheritance.
-Uses Relationship 1:1 and 1:M relationship
-Unit Tests Writing simple unit tests, White and Black box tests
Patterns Overview of the families of patterns (e.g. architectural, behavioural, analysis, and design). Implementation of sample individual patterns within the various categories.
Refactoring Principles and practice of refactoring. Refactoring and design, with specific attention to refactoring to patterns.

Module Assessment

Assessment Breakdown	%
Course Work	100.00%

Module Special Regulation

Assessments

Full Time On Campus

Course Work

Assessment Type	Continuous Assessment	**% of Total Mark**	30
Marks Out Of	0	**Pass Mark**	0
Timing	Week 7	**Learning Outcome**	1,2,3
Duration in minutes	0		

Assessment Description
Students will be given a sample project and will be required to analyse, design, code and test a solution to the problem. The assessment criteria will be concerned with the students' demonstrated understanding of and ability to apply basic object oriented techniques in the solution of given problems and a demonstrated ability to test their solutions.

Assessment Type	Class Test	**% of Total Mark**	40
Marks Out Of	100	**Pass Mark**	40
Timing	Week 9	**Learning Outcome**	1,2,3
Duration in minutes	120		

Assessment Description

Assessment Type	Continuous Assessment	**% of Total Mark**	30
Marks Out Of	0	**Pass Mark**	0
Timing	Sem 2 End	**Learning Outcome**	1,2,3,4,5
Duration in minutes	0		

Assessment Description
Students will be given a sample solution to a specified problem. They will be required to refactor this system using appropriate patterns. They will also be required to test their solution using unit testing and document the solution using UML specification diagrams.

No Project

No Practical

No Final Examination

Part Time On Campus

Course Work

Assessment Type	Continuous Assessment	**% of Total Mark**	30
Marks Out Of	0	**Pass Mark**	0
Timing	Week 7	**Learning Outcome**	1,2,3
Duration in minutes	0		

Assessment Description
Students will be given a sample project and will be required to analyse, design, code and test a solution to the problem. The assessment criteria will be concerned with the students' demonstrated understanding of and ability to apply basic object oriented techniques in the solution of given problems and a demonstrated ability to test their solutions.

Assessment Type	Class Test	**% of Total Mark**	40
Marks Out Of	0	**Pass Mark**	0
Timing	Week 9	**Learning Outcome**	1,2,3
Duration in minutes	120		

Assessment Description

Assessment Type	Continuous Assessment	**% of Total Mark**	30
Marks Out Of	0	**Pass Mark**	0
Timing	Sem 2 End	**Learning Outcome**	1,2,3,4,5
Duration in minutes	0		

Assessment Description
Students will be given a sample system to a specified problem. They will be required to refactor this system using appropriate patterns. They will also be required to test their solution using unit testing and document the solution using UML specification diagrams.

No Project

No Practical

No Final Examination

Reassessment Requirement
No repeat examination *Reassessment of this module will be offered solely on the basis of coursework and a repeat examination will not be offered.*

Module Workload

Workload: Full Time On Campus

Workload Type	*Contact Type*	*Workload Description*	*Frequency*	*Average Weekly Learner Workload*	*Hours*
Practical	Contact	No Description	Every Week	5.00	5
Tutorial	Contact		Every Week	1.00	1
Directed Reading	Non Contact	No Description	Every Week	4.00	4
Independent Study	Non Contact	No Description	Every Week	6.00	6
				Total Weekly Learner Workload	16.00
				Total Weekly Contact Hours	6.00

Workload: Part Time On Campus

Workload Type	*Contact Type*	*Workload Description*	*Frequency*	*Average Weekly Learner Workload*	*Hours*
Practical	Contact	No Description	Every Week	5.00	5
Tutorial	Contact		Every Week	1.00	1
Directed Reading	Non Contact	No Description	Every Week	4.00	4
Independent Study	Non Contact	No Description	Every Week	6.00	6
				Total Weekly Learner Workload	16.00
				Total Weekly Contact Hours	6.00

Module Resources

Recommended Book Resources

Nathan Clark. (2018), C#: Advanced Features and Programming Techniques (Step-By-Step C#) (Volume 3), CreateSpace Independent Publishing Platform, [ISBN: 978-19841214].

Sharp, John. (2013), Microsoft Visual C# 2013 Step by Step, 1st ed.. Microsoft Press, p.824, [ISBN: 978-07356818].

Greene, Jennifer; Stellman, Andrew. (2012), Head First C#, 3rd ed.. O'Reilly Media Formats, p.1100, [ISBN: 978-1-4493-43].

McLaughlin, B et al.. (2006), Head First Object-Oriented Analysis and Design, O'Reilly Media, [ISBN: 978-0596008673].

Supplementary Book Resources

Hilyard, Jay; Toilet, Stephen. (2015), C# 6.0 Cookbook, 1st. O'Reilly Media Formats, [ISBN: 978-1-4919-21].

This module does not have any article/paper resources

Other Resources

Website, C# Station,
http://www.csharp-station.com/

Website, C# Corner,
http://www.c-sharpcorner.com/

Website, Code Project,
http://www.codeproject.com/