APPROVED

SWRE I8001: Software Engineering

Module Details

Module Code:	SWRE I8001
Full Title:	Software Engineering APPROVED
Valid From::	Semester 1 - 2019/20 (June 2019)
Language of Instruction:	English

Duration:	2 Semesters

Credits::	10

Module Owner::	Michelle Graham

Departments:	Unknown

Module Description:	Students completing this module will be able to select and implement appropriate designs to solve various problems using software engineering principles.

Module Learning Outcome	
On successful completion of this module the learner will be able to:	
#	**Module Learning Outcome Description**
MLO1	Discuss and apply the principles of Software Engineering.
MLO2	Manage the quality of the software development product.
MLO3	Identify and implement appropriate Design Patterns for a particular problem.
MLO4	Specify and implement the appropriate data structures and algorithms for a range of problems.
MLO5	Analyse the space and time efficiency of selected algorithms and data structures.
MLO6	Develop recursive design and implementation 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
Software Engineering as a profession. Evolution, principles and practices.
Software Quality Quality Models, Quality Plan, Metrics, Refactoring, Design Principles
Design Patterns Common Design partterns including Singleton, Facade, Observer ,Proxy etc. UML
Data Structures Specification, application and implementation
Algorithm Design and Analysis Design and implementation of common searching and sorting algorithms.
Efficiency analysis Time and space analysis
Recursion Recursive design and implementation

Module Assessment

Assessment Breakdown	%
Course Work	60.00%
Final Examination	40.00%

Module Special Regulation

Assessments

Full Time On Campus

Course Work

Assessment Type	Written Report	**% of Total Mark**	10
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	1,2
Duration in minutes	0		

Assessment Description
Students will prepare a technical report on a software engineering topic.

Assessment Type	Class Test	**% of Total Mark**	10
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	3
Duration in minutes	0		

Assessment Description
Implement Design Patterns.

Assessment Type	Class Test	**% of Total Mark**	10
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	4,6
Duration in minutes	0		

Assessment Description
Implement data structures and algorithms

Assessment Type	Project	**% of Total Mark**	30
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	2,4,6
Duration in minutes	0		

Assessment Description
Using a case study students will specify non functional requirements and implement a partial solution to same.

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**	1,2,4,5
Duration in minutes	120		

Assessment Description
Writen exam covering theory from all aspects of the course.

Part Time On Campus

Course Work

Assessment Type	Written Report	**% of Total Mark**	10
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	1,2
Duration in minutes	0		

Assessment Description
Students will prepare a technical report on a software engineering topic.

Assessment Type	Class Test	**% of Total Mark**	10
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	3
Duration in minutes	0		

Assessment Description
Implement Design Patterns.

Assessment Type	Class Test	**% of Total Mark**	10
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	4,6

Duration in minutes	0		

Assessment Description
Implement data structures and algorithms

Assessment Type	Project	**% of Total Mark**	30
Marks Out Of	0	**Pass Mark**	0
Timing	n/a	**Learning Outcome**	2,4,6
Duration in minutes	0		

Assessment Description
Using a case study students will specify non functional requirements and implement a partial solution to same.

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**	1,2,4,5
Duration in minutes	120		

Assessment Description
Writen exam covering theory from all aspects of the course.

Reassessment Requirement

A repeat examination
Reassessment of this module will consist of a repeat examination. It is possible that there will also be a requirement to be reassessed in a coursework element.

Module Workload

Workload: Full Time On Campus					
Workload Type	*Contact Type*	*Workload Description*	*Frequency*	*Average Weekly Learner Workload*	*Hours*
Lecturer-Supervised Learning (Contact)	Contact	No Description	Every Week	1.00	1
Practical	Contact	No Description	Every Week	2.00	2
Directed Reading	Non Contact	No Description	Every Week	2.00	2
Independent Study	Non Contact	No Description	Every Week	3.00	3
				Total Weekly Learner Workload	8.00
				Total Weekly Contact Hours	3.00

Workload: Part Time On Campus					
Workload Type	*Contact Type*	*Workload Description*	*Frequency*	*Average Weekly Learner Workload*	*Hours*
Practical	Contact	No Description	Every Week	2.00	2
Lecturer-Supervised Learning (Contact)	Contact	No Description	Every Week	1.00	1
Directed Reading	Non Contact	No Description	Every Week	2.00	2
Independent Study	Non Contact	No Description	Every Week	3.00	3
				Total Weekly Learner Workload	8.00
				Total Weekly Contact Hours	3.00

Module Resources

Recommended Book Resources

Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. (2014), Data Structures and Algorithms in Java, 6th Edition International Student Version. John Wiley & Sons, Inc., p.720, [ISBN: 978-1-118-808].

Eric Freeman, Elisabeth Robson, Bert Bates & Kathy Sierra. (2014), Head First Design Patterns, 2nd edition. O'Reilly Media, p.694, [ISBN: 0-596-00712-4].

Sommerville. Software Engineering, 10th Edition.

Martin Fowler. Refactoring: Improving the Design of Existing Code,.

Supplementary Book Resources

Dale, Joyce, Weems. (2016), Object Oriented Data Structures using Java, 4th. [ISBN: 978-128408909].

Gary McLean Hall. (2017), Adaptive Code: Agile coding with design patterns and SOLID principles, 2nd. Microsoft Press, [ISBN: 978-150930258].

Kathy Sierra & Bert Bates. (2005), Head First Java, 2nd. O'Reilly Media, p.720, [ISBN: 0-596-00920-8].

This module does not have any article/paper resources

This module does not have any other resources