

**COMP I8008: Concurrent & Distributed
Programming**

Module Details	
Module Code:	COMP I8008
Full Title:	Concurrent & Distributed Programming APPROVED
Valid From::	Semester 1 - 2019/20 (June 2019)
Language of Instruction:	English
Duration:	1 Semester
Credits::	5
Module Owner::	Michelle Graham
Departments:	Unknown
Module Description:	The class focusses on issues that arise in the design and implementation of concurrent and distributed applications. (1,2,4,6)

Module Learning Outcome	
On successful completion of this module the learner will be able to:	
#	Module Learning Outcome Description
MLO1	Design and implement concurrent programs
MLO2	Design solutions to synchronization issues within concurrent programs through appropriate use of locks, semaphores and monitors.
MLO3	Design and implement distributed programs using current middleware technologies
MLO4	Analyse and evaluate the fundamental concepts underlying distributed programming including message passing, remote method invocation and web services.
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 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).</i>	
No recommendations listed	

Module Indicative Content
Multi-threaded/Concurrent Programming Process and Synchronization, Locks and Barriers, Semaphores, Monitors, parallelization using stream/functional approach
Distributed Systems Goals of a distributed system, Hardware concepts, Software concepts, safe access to shared distributed resources
Client-Server Applications Clients and Servers Architectures, Pooling vs Thread-Per-Client, Application Layering, Client-Server Architectures, Non-text-based servers
Middleware technologies TCP/IP Sockets, RPC/RMI, Web Services

Module Assessment

Assessment Breakdown	%
Course Work	50.00%
Final Examination	50.00%

Module Special Regulation

Assessments

Full Time On Campus

Course Work			
Assessment Type	Project	% of Total Mark	25
Marks Out Of	0	Pass Mark	0
Timing	n/a	Learning Outcome	3,4
Duration in minutes	0		
Assessment Description Distributed application development focusing on appropriate underlying distributed system technology (e.g. message passing, web services, distributed objects etc)			

Project			
Assessment Type	Project	% of Total Mark	25
Marks Out Of	0	Pass Mark	0
Timing	n/a	Learning Outcome	1,2
Duration in minutes	0		
Assessment Description Concurrent programming project demonstrating understanding of management of resources			

No Practical

Final Examination			
Assessment Type	Formal Exam	% of Total Mark	50
Marks Out Of	0	Pass Mark	0
Timing	End-of-Semester	Learning Outcome	1,2,3,4
Duration in minutes	120		
Assessment Description Formal exam covering theory from all aspects of the course			

Part Time On Campus

Course Work			
Assessment Type	Project	% of Total Mark	25
Marks Out Of	0	Pass Mark	0
Timing	n/a	Learning Outcome	3,4
Duration in minutes	0		
Assessment Description Distributed application development focusing on appropriate underlying distributed system technology (e.g. message passing, web services, distributed objects etc)			

Project			
Assessment Type	Project	% of Total Mark	25
Marks Out Of	0	Pass Mark	0
Timing	n/a	Learning Outcome	1,2
Duration in minutes	0		
Assessment Description Concurrent programming project demonstrating understanding of management of resources			

No Practical

Final Examination			
Assessment Type	Formal Exam	% of Total Mark	50
Marks Out Of	0	Pass Mark	0
Timing	End-of-Semester	Learning Outcome	1,2,3,4
Duration in minutes	0		
Assessment Description Formal exam covering theory from all aspects of the course			

Reassessment Requirement			
A repeat examination <i>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.</i>			

Module Workload

Workload: Full Time On Campus

Workload Type	Contact Type	Workload Description	Frequency	Average Weekly Learner Workload	Hours
Lecturer-Supervised Learning (Contact)	Contact	3 lab-based classes per week integrating theory and practical content. Classes will normally be broken down into 50% theory delivery and 50% practical implementation.	Every Week	3.00	3
Directed Reading	Non Contact	Consumption of lecturer-recommended content (videos /papers/book content/web articles)	Every Week	2.00	2
Independent Study	Non Contact	Independent practical work	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
Directed Reading	Non Contact	Consumption of lecturer-recommended content (videos /papers/book content/web articles)	Every Week	2.00	2
Independent Study	Non Contact	Independent practical work	Every Week	4.00	4
Lecturer Supervised Learning	Contact	lab-based classes per week integrating theory and practical content. Classes will normally be broken down into 50% theory delivery and 50% practical implementation.	Every Week	2.00	2
Total Weekly Learner Workload					8.00
Total Weekly Contact Hours					2.00

Module Resources

Recommended Book Resources

Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes and Doug Lea. (2006), Java Concurrency in Practice, 1st edition. AddisonWesley, [ISBN: 0321349601].

Andrew S Tanenbaum and Maarten Van Steen. (2013), Distributed Systems: Principles and Paradigms, 2nd edition. Pearson, p.640, [ISBN: 1292025522].

Supplementary Book Resources

Kathy Sierra & Bert Bates. (2005), Head First Java, 2nd edition. 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