



Title:	Distributed Sys. Programming APPROVED	
Long Title:	Distributed Sys. Programming	
Module Code:	SOFT8023	
Duration:	1 Semester	
Credits:	5	
NFQ Level:	Advanced	
Field of Study:	Computer Software	
Valid From:	Semester 1 - 2017/18 (September 2017)	
Module Delivered in	3 programme(s)	
Module Coordinator:	Sean McSweeney	
Module Author:	DENIS LONG	
Module Description:	In a distributed system computers are connected through a network and co-ordinate with each other with the aim of achieving a common goal or task. In these systems, processes typically communicate and synchronise by means of messages. In this module, students will learn how to program distributed systems and introduces them to multi-threading and synchronisation. In addition, the module will examine the main communication mechanisms used between processes using client-servers, remote objects and procedure calls.	
Learning Outcomes		
<i>On successful completion of this module the learner will be able to:</i>		
LO1	Evaluate and apply design patterns in the design and development of a distributed system.	
LO2	Assess and apply different architectural patterns in a distributed system.	
LO3	Critically access and apply threading in a distributed application.	
LO4	Debug a distributed client/server application, identifying object properties and variables at run-time.	
LO5	Create a distributed object application using RMI, allowing client/server to communicate securely via interfaces and objects.	
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
12793	COMP7013	Object Oriented Programming
12805	SOFT8023	Distributed Sys. Programming
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

Distributed Design Patterns

What is a distributed design pattern? Patterns related to distributed computing e.g Proxy, Adapter, single thread execution.

Client/Server Programming

Server Socket; reading from and writing to a socket; Developing a multi-threaded server; TCP and UDP client/server.

Multi-Threading

Thread States; life cycle of a thread; thread Priorities and thread scheduling; thread synchronization. Critical sections, Asynchronous programming.

Distributed Objects

Remote Method Call; Distributed object systems; Local surrogate object; Invocation of methods on a remote object;

Debugging and Testing

Debugging in a distributed environment; Testing that the distributed system actually works.

Framework Services for distributed objects

Garbage collection; Registry services; Signing objects (so that you know that what was signed is what is retrieved); Sealing an object (so that only the intended recipients can view the object).

Assessment Breakdown	%
Course Work	50.00%
End of Module Formal Examination	50.00%

Course Work

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Project	An example assessment would be to create a simple client server application using sockets.	1,2,4	20.0	Week 6
Project	Programming assignment(s) using the technologies covered in the lectures. Example assignment(s) will access if a student can apply design patterns to write distributed code, use technologies such as RMI and secure communication and information in transit.	1,2,3,4,5	30.0	Week 12

End of Module Formal Examination

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Formal Exam	End of Semester Formal Examination.	1,2,3,4,5	50.0	End-of-Semester

Reassessment Requirement

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.

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	Lectures delivering the theory underpinning the learning outcomes.	2.0	Every Week	2.00
Lab	Lab supporting the learning outcomes and content delivered in lectures.	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	Lectures delivering the theory underpinning the learning outcomes.	2.0	Every Week	2.00
Lab	Lab supporting the learning outcomes and content delivered in lectures.	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

Recommended Book Resources

- Cay S. Horstmann , Gary Cornell 2013, *Core Java: Advanced Features Volume II*, 2 Ed., all, Prentice Hall [ISBN: 9780134177298]

Supplementary Book Resources

- Paul Deitel, Harvey Deitel, 2014, *Java How to Program*, 10 Ed., Pearson [ISBN: 9780132575652]
- Uttam Kumar Roy 2015, *Advanced Java Programming*, 1 Ed., Oxford University Press [ISBN: 9780199455508]
- Brian Goetz 2006, *java concurrency in practice*, 1 Ed., Addison-Wesley Professional us [ISBN: 9780321349606]
- Eric Freeman, Elisabeth Freeman; with Kathy Sierra, Bert Bates 2004, *Head first design patterns*, 1 Ed., O'Reilly Media [ISBN: 9780596007126]

Recommended Article/Paper Resources

- Oracle 2016, *RMI*
<https://docs.oracle.com/javase/tutorial/rmi/>
- Tutorialpoint 2016, *Design patterns*
https://www.tutorialspoint.com/design_pattern/

This module does not have any other resources

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	<u>Bachelor of Science (Honours) in Software Development</u>	5	Mandatory
CR_KDNET_8	<u>Bachelor of Science (Honours) in Computer Systems</u>	5	Mandatory
CR_KCOMP_7	<u>Bachelor of Science in Software Development</u>	5	Mandatory