GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

COMP08027 Advanced Object Oriented Software Development / Project

Full Title	Advanced Object Oriented Software Development / Project		
Status	Uploaded to Banner	**Start Term**	2012
NFQ Level	08	**ECTS Credits**	10
Module Code	COMP08027	**Duration**	13 weeks - (13 Weeks)
Grading Mode		**Department**	Comp Science & Applied Physics
Module Author	Dr. John Healy		

Module Description

This module covers advanced object-oriented concepts such as threads, abstraction, encapsulation, inheritance and polymorphism. In addition, common data structures, algorithms and the Java Collections Framework are studied. The assessment of this 10-credit module includes a capstone project worth 50% of the total credits.

This project is unique in that it provides a pedagogic scaffold for learners through its time-tabled and lecture supported delivery, whilst still providing significant (and necessary) self-directed learning opportunities through the appropriately weighted capstone project.

The project aims to encompass and embrace the totality of the modules in the programme, with specific emphasis on software design and software engineering, underpinned by core cloud computing, networking and database paradigms.

	Learning Outcomes ***On completion of this module the learner will/should be able to:***
1.	Implement advanced OO features in software applications
2.	Design and implement concurrency control in a Java application.
3.	Use well-known algorithms and data structures in a software solution
4.	Implement algorithms and data structures using the Java programming language
5.	Implement a capstone project which demonstrates the integration of many or all of the constituent components (where appropriate) of the programme with specific emphasis on software design and engineering and underpinned by core Cloud computing, Networking and Database paradigms.

Indicative Syllabus

Advanced OO programming (50%): Abstraction, encapsulation, inheritance and polymorphism. Abstract classes and interfaces. Design principles and a sample of important design patterns. Serialization, persistence; The Java Collections Framework and generics.

Concurrency (15%): Designing concurrent applications, threading

Data Structures & Algorithms (35%): Data structures and algorithms and their implementation in an OO language e.g. analysis of algorithms lists, queues, trees and graphs.

Teaching and Learning Strategy

Assessment Strategy

Repeat Assessment Strategies

Indicative Coursework and Continuous Assessment:		100 %		
Form	***Title***	***Percent***	***Week (Indicative)***	***Learning Outcomes***
UNKNOWN	Practical Evaluation Assessment of knowledge of advanced object-oriented concepts.	50 %	OnGoing	1,2,3,4
UNKNOWN	Project demonstrating consolidation and integration of learning from other modules in programme	50 %	End of Term	5

Full Time Delivery Mode Average Weekly Workload:			6.00 Hours		
Type	***Description***	***Location***	***Hours***	***Frequency***	***Weekly Avg***
Practical	Programming Practical	Computer Laboratory	6	Weekly	6.00

Literary Resources

- *Objects, Abstractions, Data Structures & Design Using Java*, Koffman, Wiley, 2005. ISBN: 0-471-46756-1
- *Head First Java, 2nd Edition*, K. Sierra, B. Bates, O'Reilly Media, 2005, ISBN: 059-6009208

Other Resources

None

Additional Information

None

Programme Membership

GA_KSOFG_L08 201700 Higher Diploma in Science in Software Development