MODULE	OBJECT ORIENTED DEVELOPMENT (course code BSCH-OOD)

Griffith College Dublin – Computing Science

Fall semester

This module is intended for at least sophomore level students who are majoring in this field (not intended for first-year students).

Aims

This module introduces you to the fundamental concepts of object oriented program design and how to use modelling for constructing complex software systems.

Learning Outcomes

Upon successful completion of this module, you should be able to:

1. develop the analytical skills necessary to apply abstract concepts in an object oriented manner
2. express system solutions in a formal manner and implement the derived formalisation
3. develop confidence in and awareness of the capabilities of object oriented development
4. produce correct software designs
5. analyse a problem, produce high quality software designs using Universal Modelling Language (UML) notation and relate the software designs to the implementation
6. identify problems associated with traditional methods of software specification, and explain how formal methods overcome these problems
7. develop high quality software that is reliable, reusable and maintainable

Indicative Content

Topic	Description
Introduction and motivation	Review of procedural paradigm, control structures and data types with emphasis on structured data types and array processing. Case studies of software engineering failures.
The object paradigm	Classification: Objects and Object Types (Classes); Abstraction; Encapsulation: Data and Behaviour; Information Hiding: Access Specifiers; Inheritance and Polymorphism; Aggregation and Association; Software Reuse;
Unified modelling language	Rationale and history of UML; Use Case Analysis; Structural View: Class and Object diagrams; Behaviour View: Sequence, Collaboration, Statechart, Activity diagrams; Environment View;
Object oriented programming	Implementation of classes and objects; Static and Dynamic Objects; Testing and debugging in Java; Use of commercial libraries; Sample programs;
Object oriented design	Implementation Options; Object Oriented Methodologies; Use of iterative development; Introduction to Patterns and Anti-Patterns
Issues in OOD	Pros and cons of the OO approach; Aspect-Oriented programming; Object-Oriented Databases; Remote/Distributed Objects