

MODULE: Object Oriented Design & Implementation

SEMESTER	II (Spring)
STAGE	II
Number of Credits:	4 semester credits/6 quarter units
THEMES	Software Development
ASSESSMENT	Continuous Assessment 50%
	Examination 50%

Aims

This module introduces students to the fundamental concepts of object oriented program design and how to use the Object Modelling paradigm for constructing software systems from requirement specifications.

Learning Outcomes

Upon completion of this module, a student will be able to:

- Select and apply appropriate object oriented design techniques to system solutions.
- Apply abstract concepts using an object oriented programming approach.
- Express system solutions in a formal manner and implement the derived formalisation.
- Develop confidence in and awareness of the capabilities of object oriented development.
- Demonstrate an ability to abstract problem specifications and program design by producing good software designs.
- Analyse problems and produce appropriate high quality software designs using Universal Modelling Language (UML) notation and relate the software designs to the implementation.
- Identify problems associated with traditional methods of software specification, and explain how formal methods overcome these problems.
- Develop high quality software that is reliable, reusable and maintainable.

Indicative Content

Topic	Description
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;
Methods for Object Oriented Analysis	Object Behaviour Analysis Object Modelling Techniques Information Modelling Classes, Associations and Attributes Object Relationship Diagrams Dynamic Modelling States, Events Transitions Scenarios and Events Traces State Transition diagrams Functional Modelling Activity Diagrams
Object Oriented Design	Implementation Options; Object Oriented Methodologies; Use of iterative development; Introduction to Patterns and Frameworks;
Object Oriented Programming	Implementation of classes and objects; Static and Dynamic Objects; Testing and debugging in Java; Use of commercial libraries; Sample programs;
Evaluation of the OOD approach	The management perspective Pros and cons of the OO approach Changeover methods and difficulties Suitable application categories Current Issues Standardisation: OMG, COBRA, etc. OO-DBMS v RDBMS OO – OS

Teaching and Learning Methods

The module will be taught using a combination of lectures, demonstrations and tutorials, practical work using Object Oriented languages, such as Java, library and other research, study of texts and handouts and interactive teaching and a case study.

Assessment Methods

Assessment will include both a continuous component and an end of semester examination. The continuous assessment component is used to develop practical skills of programming techniques and will be based both on the lab workbooks and graded assignments / in class tests. Students will be expected to develop efficient, well-documented code, meeting accepted quality standards.

Primary Reading List

Title	Author	Publisher
Thinking in Java 3 rd Edition	Eckel B.	Prentice Hall
Learning Java 2 nd Edition	Niemeyer P. Knudson J.	O'Reilly
A First Course in Programming Java	Mullins T.	Mullins

Recommended Reading List

Object-oriented Software Construction	Meyer B.	Prentice Hall
Object-Oriented Program Design	Mullin M.	Addison-Wesley
Java Simplified	Shaw A.	A.D.R Limited
Java – How to Program 5 th Edition	Deitel H. Deitel P.	Prentice Hall