

MC5305 OBJECT ORIENTED ANALYSIS AND DESIGN

DETAILED SYLLABUS

OBJECTIVES:

- To provide a brief, hands-on overview of object-oriented concepts and its life cycle for software development.
- To learn for modelling the software and to design them using UML diagrams
- To understand the problem domain and to identify the objects from the problem specification.
- To understand, how to apply design axioms and corollaries for the classes and object relational systems.
- To gain knowledge about open source tools for Computer Aided Software Engineering

UNIT I INTRODUCTION

An overview – Object basics – Object state and properties – Behaviour – Methods – Messages – Information hiding – Class hierarchy – Relationships – Associations – Aggregations- Identity – Dynamic binding – Persistence – Meta classes – Object oriented system development life cycle.

UNIT II METHODOLOGY AND UML

Introduction – Survey – Rumbaugh, Booch, Jacobson methods – Unified modelling language – Static and Dynamic models – Rational Rose Suite - UML diagrams – Static diagram: Class diagram – Use case diagrams – Behaviour Diagram: Interaction diagram – State chart diagram – Activity diagram - Implementation diagram: Component diagram – Deployment diagram – example - Design of online railway reservation system using UML diagrams - Dynamic modelling – Model organization – Extensibility.

UNIT III OBJECT ORIENTED ANALYSIS

Identifying Use case – Business object analysis – Use case driven object-oriented analysis – Use case model – Documentation – Classification – Identifying object, relationships, attributes, methods – Super-sub class – A part of relationships Identifying attributes and methods – Object responsibility – construction of class diagram for generalization, aggregation – example – vehicle class.

UNIT IV OBJECT ORIENTED DESIGN

Design process and benchmarking – Axioms – Corollaries – Designing classes – Class visibility – Refining attributes – Methods and protocols – Object storage and object interoperability – Databases – Object relational systems – Designing interface objects – Macro and Micro level processes – The purpose of a view layer interface-OOUI – MVC Architectural Pattern and Design – Designing the system.

UNIT V CASE TOOLS

Railway domain: Platform assignment system for the trains in a railway station – Academic domain: Student Marks Analysing System - ATM system - Stock maintenance – Quiz System - E-mail Client system - Cryptanalysis – Health Care Systems. Use Open source CASE Tools: StarUML/ UML Graph for the above case studies.

REFERENCES

1. Ali Bahrami, "Object Oriented System Development", McGraw Hill International Edition, 2008
2. Brahma Dathan, Sarnath Ramnath, "Object-Oriented Analysis, Design and Implementation", Universities Press, 2010
3. Bernd Bruegge, Allen H. Dutoit, Object Oriented Software Engineering using UML, Patterns and Java, Pearson 2004
4. Craig Larman, Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and Iterative Development", 3rd Edition, Pearson Education, 2005
5. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Addison Wesley Long man, 1999
6. Martin Fowler, "UML Distilled A Brief Guide to Standard Object Modeling Language", 3rd Edition, Addison Wesley, 2003
7. Russ Miles, Kim Hamilton, "Learning UML 2.0", O'Reilly, 2008
8. http://staruml.sourceforge.net/docs/StarUML_5.0_Developer_Guide.pdf
9. <http://www.spinellis.gr/umlgraph/doc/index.html>