

OBJECTIVES:

The student should be made to:

- Learn the basics of OO analysis and design skills.
- Learn the UML design diagrams.
- Learn to map design to code.
- Be exposed to the various testing techniques.

UNIT I **UML DIAGRAMS** **9**

Introduction to OOAD – Unified Process – UML diagrams – Use Case – Class Diagrams – Interaction Diagrams – State Diagrams – Activity Diagrams – Package, component and Deployment Diagrams.

UNIT II **DESIGN PATTERNS** **9**

GRASP: Designing objects with responsibilities – Creator – Information expert – Low Coupling – High Cohesion – Controller – Design Patterns – creational – factory method – structural – Bridge – Adapter – behavioral – Strategy – observer.

UNIT III **CASE STUDY** **9**

Case study – the Next Gen POS system, Inception -Use case Modeling – Relating Use cases – include, extend and generalization – Elaboration – Domain Models – Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies – Aggregation and Composition.

UNIT IV **APPLYING DESIGN PATTERNS** **9**

System sequence diagrams – Relationship between sequence diagrams and use cases Logical architecture and UML package diagram – Logical architecture refinement – UML class diagrams – UML interaction diagrams – Applying GoF design patterns.

UNIT V **CODING AND TESTING** **9**

Mapping design to code – Testing: Issues in OO Testing – Class Testing – OO Integration Testing – GUI Testing – OO System Testing.

TOTAL: 45 PERIODS

OUTCOMES:

At the end of the course, the student should be able to:

- Design and implement projects using OO concepts.
- Use the UML analysis and design diagrams.
- Apply appropriate design patterns.
- Create code from design.
- Compare and contrast various testing techniques.

TEXT BOOK:

1. Craig Larman, “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”, Third Edition, Pearson Education, 2005.

REFERENCES:

1. Simon Bennett, Steve Mc Robb and Ray Farmer, “Object Oriented Systems Analysis and Design Using UML”, Fourth Edition, Mc-Graw Hill Education, 2010.
2. Erich Gamma, a n d Richard Helm, Ralph Johnson, John Vlissides, “Design patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995.
3. Martin Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, Third edition, Addison Wesley, 2003.
4. Paul C. Jorgensen, “Software Testing:- A Craftsman’s Approach”, Third Edition, Auerbach Publications, Taylor and Francis Group, 2008.