| COURSE CODE | COURSE TITLE | L | T | P | C |
|---|---|---|---|---|---|
| UCS1604 | OBJECT ORIENTED ANALYSIS AND DESIGN | 3 | 0 | 0 | 3 |

## OBJECTIVES

- To understand and differentiate Unified Process from other approaches
- To understand object-oriented software design using UML's static diagrams
- To understand software modeling using the UML's dynamic diagrams
- To learn improving software design with design patterns
- To learn testing the software with its requirements specification.

## UNIT I DEVELOPMENT PROCESS & USE-CASE DIAGRAM 9

Introduction to OOAD with OO Basics -- Unified process -- UML diagrams -- Use case -- Case study -- The Next Gen POS system, Inception – Use case modelling -- Relating use cases -- Include, extend and generalization -- When to use use-cases.

## UNIT II UNIFIED PROCESS & CLASS DIAGRAM 9

Class diagram -- Elaboration -- Domain model -- Finding conceptual classes and description classes -- Associations -- Attributes -- Domain model refinement -- Finding conceptual class hierarchies -- Aggregation and composition -- Relationship between sequence diagrams and use cases -- When to use class diagrams.

## UNIT III DYNAMIC & IMPLEMENTATION DIAGRAMS 9

Dynamic Diagrams: UML interaction diagrams -- System sequence diagram -- Collaboration diagram -- When to use communication diagrams -- State machine diagram and modelling -- When to use state diagrams -- Activity diagram -- When to use activity diagrams. Implementation Diagrams: UML package diagram -- When to use package diagrams -- Component and deployment diagrams -- When to use component and deployment diagrams.

## UNIT IV DESIGN PATTERNS 9

Designing objects with responsibilities -- Creator – Information expert -- Low coupling -- High cohesion -- Controller design patterns -- Creational -- Factory method -- Structural -- Bridge -- Adapter -- Behavioural -- Strategy -- Observer -- Applying GoF design patterns -- Mapping design to code.

## UNIT V TEST DRIVEN DEVELOPMENT AND REFACTORING 9

Object oriented methodologies -- Software quality assurance – Impact of object orientation on testing -- Develop test cases and test plans.

**TOTAL PERIODS: 45**

## OUTCOMES

**On successful completion of this course, the student will be able to**

- Express software design with UML diagrams (K2)
- Design and implement projects using OO concepts (K4)
- Identify and map basic software requirements in UML mapping (K3)

- Transform UML based software design into pattern based design using design patterns (K4)
- Test any object-oriented software against its requirements (K3).

**TEXTBOOKS**

1. Larman, Craig, “Applying UML and Patterns”, Pearson Education Asia,2008.
2. Ali Bahrami, “Object Oriented Systems Development”, McGraw Hill International Edition,1999.

**REFERENCE BOOKS**

1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, “Design patterns: Elements of Reusable Object-Oriented Software”, Addison Wesley, 1995.
2. Martin Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, 3rd edition, Addison Wesley, 2003.
3. Booch, G, Jacobson I, Rumbaugh J, “The Unified Modeling Language UserGuide”, Addison Wesley, 2008.
4. Roger S Pressman, “Software Engineering – A Practitioner ’s Approach”,7thedition, 2010.
5. Aditya P Mathur, “Foundations of SoftwareTesting– FundamentalAlgorithms andTechniques”, Dorling Kindersley (India)Pvt.Ltd., Pearson Education, 2008.