

**Unit – I Software Engineering:**

- ❖ Introduction;
- ❖ Software Life Cycle Models;
- ❖ Object Oriented Concepts and Modeling –
  - Introduction to Class,
  - Object,
  - Inheritance,
  - Polymorphism;
- ❖ Overview of System Analysis and Design.
- ❖ Requirement Analysis:
  - Introduction to software specification,
  - its needs and importance,
  - formal specification methods,
- ❖ SRS: attributes of good SRS and organization of SRS document.
  
- ❖ Introduction to Unified Modelling Language (UML);
  - Diagrams - Notations

## **Unit – II Object Methodology & Requirement Elicitation:**

### **Introduction to Object Oriented Methodology,**

- Overview of Requirements Elicitation,
- Requirements Elicitation Activities,
- Managing Requirements Elicitation.

### **Requirement Engineering:**

- Introduction;
- Requirement Engineering Tasks and Process;
- Analysis - An Overview of Analysis,
- Analysis Object Models and Dynamic Models;
- Process Models and Design Models;

### **Software Engineering Models:**

- ❖ Basic structural Modeling –
- ❖ Classes, Relationships,
- ❖ Common mechanisms,
- ❖ Diagrams,
- ❖ Class diagrams;
- ❖ Advanced Structural Modeling –
  - Interfaces,
  - types and Roles Object diagrams,
  - packages.

### **Unit – III System Design:**

- ❖ An Overview of System Design,
- ❖ System Design Activities;
- ❖ Identifying Design Goals;
- ❖ Documenting System Design and Reviewing System Design.

### **Basic Behavioral Modeling:**

- ❖ Use cases,
- ❖ use case diagrams,
- ❖ Interaction diagram,
- ❖ Activity diagrams,
- ❖ statechart diagrams,
- ❖ component diagrams,
- ❖ deployment diagrams,
- ❖ Sequence Diagram,
- ❖ patterns and frame works.

### **A Case Study:**

(Example: - ATM, Trading System, Banking System, Library Information System, Student Information System etc.).

## **Unit – IV Object Oriented Analysis and Design:**

- ❖ Analysis and design tools –
- ❖ E-R analysis,
- ❖ Decision tree and decision tables,
- ❖ DFD (physical and logical),
- ❖ Data dictionary-definition,
- ❖ component,
- ❖ advantages;
- ❖ Input and output design;
- ❖ Components of OO Analysis Models;
- ❖ OOA Process;
- ❖ System Design and Object Design Process;

## **Software Testing:**

- ❖ Testing Fundamentals,
- ❖ White Box Testing,
- ❖ Black Box Testing,
- ❖ software testing strategies,
- ❖ verification and Validation,
- ❖ System Testing,
- ❖ Unit testing,
- ❖ Integration testing and Debugging.
- ❖ Software Maintenance – Maintenance Tasks,
- ❖ Characteristics of a good quality software.
- ❖ Object Oriented Testing:
- ❖ View of Testing;
- ❖ Testing OOA and OOD Models;
- ❖ OOT Strategies;
- ❖ Test case design for OO Software;
- ❖ Testing Method;
  - Inter class test case design;

**Unit – V Metrics and Quality:**

- ❖ Introduction;
- ❖ Software Quality;
- ❖ Metrics - project based metrics and design based metrics,
- ❖ Process, Products, Resources;
- ❖ Measuring quality ; GQM;

**Quality of OOD:**

- ❖ Principles of OOD - General Principles,
- ❖ Cohesion Principles,
- ❖ Coupling Principles.
- ❖ Metrics for OO Design - Metrics Design Model,

**MOOD Metrics Model;**

- ❖ Evaluation of OO Metrics;
- ❖ Quality Management:
- ❖ Process and Product Quality;
- ❖ Quality Assurance and Standards;
- ❖ Quality Planning; Quality Control;
- ❖ Software Measurement and Metrics;

Text Books:

- [1] Bernd Bruegge and Allen H. Dutoit, “Object-Oriented Software Engineering”, 2<sup>nd</sup> Edition, Pearson.
- [2] Ali Bahrami, “Object Oriented Systems Development”, 2<sup>nd</sup> Edition, 2012, McGraw Hill Publishers.

References:

- [1] Hans Van Vliet, “Software Engineering – Principles and Practice”, Second Edition, Vrije Universiteit, Amsterdam.
- [2] Waman S. Jawadekar, “Software Engineering – Principles and Practice”, Computer Engineering Series, Tata McGraw-Hill Publishing Company Limited.
- [3] Pankaj Jalote, “An Integrated Approach to Software Engineering”, third Edition, Narosa Publishing House.
- [4] Stephen Schach, “Classical Object Oriented Software Engineering with UML and Java”, 2008, McGraw-Hill.
- [5] Graddy Booch, James Rumbaugh, Ivar Jacobson, “The Unified Modeling Language”, 3<sup>rd</sup> Edition, 2009, Pearson.