SAVITRIBAI PHULE PUNE UNIVERSITY
T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS
TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16
TITLE OF PAPER : Object Oriented Software Engineering
Code No. : CS-336

Semester III **Total Lectures: 48**

Prerequisites

- Knowledge of Object Oriented Concepts
- Knowledge of Classical Software Engineering

Aim

To Understand Object Oriented Modeling techniques and their applicability.

Objectives

- Understanding importance of Object Orientation in Software engineering
- Understand the components of Unified Modeling Language
- Understand techniques and diagrams related to structural modeling
- Understand techniques and diagrams related to behavioral modeling
- Understand techniques of Object Oriented analysis, design and testing

1. Object Oriented Concepts and Principles [4]

1.1 What is Object Orientation ? - Introduction , Object , Classes and Instance , Polymorphism, Inheritance

1. 2 Object Oriented System Development- Introduction, Function/Data Methods (With Visibility), Object Oriented Analysis, Object Oriented Construction

1.3 Identifying the Elements of an Object Model

1.4 Identifying Classes and Objects

1.5 Specifying the Attributes (With Visibility)

1.6 Defining Operations

1.7 Finalizing the Object Definition

2. Introduction to UML [2]

2.1 Concept of UML

2.2 Advantages of UML

3. Basic Structural Modeling [5]

3.1 Classes

3.2 Relationship

3.3 Common Mechanism

3.4 Class Diagram (Minimum three examples should be covered)

4. Advanced Structural Modeling [7]

4.1 Advanced Classes

4.2 Advanced Relationship

4.3 Interface

4.4 Types and Roles

4.5 Packages

4.6 Object Diagram (Minimum three examples should be covered)

5. Basic Behavioral Modeling [9]

5.1 Interactions
5.2 Use Cases and Use Case Diagram with stereo types (Minimum three examples should be covered)
5.3 Interaction Diagram (Minimum two examples should be covered)
5.4 Sequence Diagram (Minimum two examples should be covered)
5.6 Activity Diagram (Minimum two examples should be covered)
5.6 State Chart Diagram (Minimum two examples should be covered)

6. Object Oriented Analysis [6]
6.1 Iterative Development and the Rational Unified Process
6.2 Inception
6.3 Understanding Requirements
6.4 Use Case Model From Inception to Elaboration
6.5 Elaboration

7. Object Oriented Design [4]
7.1 The Booch Method, The Coad and Yourdon Method and Jacobson Method and Raumbaugh Method
7.2 The Generic Components of the OO Design Model
7.3 The System Design Process - Partitioning the Analysis Model, Concurrency and Sub System Allocation, Task Management Component, The Data Management Component, The Resource Management Component, Inter Sub System Communication
7.4 Object Design Process

8. Architectural modeling [6]
8.1 Component
8.2 Components Diagram (Minimum two examples should be covered)
8.3 Deployment Diagram (Minimum two examples should be covered)
8.4 Collaboration Diagram (Minimum two examples should be covered)

9. Object Oriented Testing [5]
9.1 Object Oriented Testing Strategies
9.2 Test Case Design for Object Oriented Software
9.3 Inter Class Test Case Design
(Use of any freeware designing tool)

References.

1. Grady Booch, James Rambaugh, The Unified Modeling Language User/Reference Guide, Pearson Education INC
2. Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC
3. Craig Larman, Applying UML and Patterns Pearson Education INC
4. Bennett, Simon, Object Oriented Analysis and Design McGraw Hill

SAVITRIBAI PHULE PUNE UNIVERSITY
T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS
TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16
TITLE OF PAPER :Computer Graphics
Code No. : CS-346

Semester IV **Total Lectures: 48**

Pre – Requisites

- Computer programming skills in C programming language
- Basic understanding of use of data structures
- Basic Mathematical concepts related to matrices and geometry

Objectives

- To study how graphics objects are represented in Computer
- To study how graphics system in a computer supports presentation of graphics information
- To study how interaction is handled in a graphics system
- To study how to manipulate graphics object by applying different transformations
- To provide the programmer's perspective of working of computer graphics

1. Introduction to Computer graphics **[4]**

1. 1 Introduction to computer graphics & graphics systems
1.2 Components of Computer Graphics Representation, Presentation , Interaction and Transformations
1.3 Applications of Computer Graphics
1.3 Pixel/Point ,Raster v/s Vector ,RGB color model, intensity
1.4 Programming essentials – event driven programming. OpenGL library

2. Input devices and Interaction tasks **[4]**

2.1 Logical Interaction – Locator, valuator , pick and choice;
2.2 Physical devices used for interaction – keyboard, mouse, trackball,spaceball, tablets, light pen, joy stick, touch panel, data glove;
2.4 Keyboard , Mouse interaction in OpenGL
2.5 Graphical User Interfaces- cursors , radio buttons, scroll bars, menus, icons
2.6 Implementing GUI in open GL

3. Presentation and Output devices ` **[4]**

3.1 Presentation Graphics - frame buffer, display file, lookup table;
3.2 Display devices, Random and Raster scan display devices; CRT,
3.3 Hardcopy devices - Plotters and Printers

4. Raster Scan Graphics **[10]**

4.1 Line drawing algorithms; DDA algorithm, Bresenham's line drawing algorithm, Circle generation algorithm;
4.2 Scan conversions- Generation of the Display, Image compression
4.3 Displaying Lines and characters
4.3 Polygon filling -Scan converting polygons, fill algorithms, Boundary fill algorithm, flood fill algorithm

5. Transformations **[7]**

5.1 Basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, Reflection, shear
5.2 Transformation of points, lines, parallel lines, intersecting lines. Viewing pipeline
5.3 Window to viewport co-ordinate transformation. Setting window and viewport in OpenGL.

6 Clipping [7]

6.1 clipping operations , point clipping ,

6.2 Line clipping; Cohen Sutherland algorithm, Midpoint subdivision algorithm, Cyrus beck algorithm;

6.3 Polygon clipping , Sutherland Hodgman algorithm, Weiler-Atherton Algorithm

7 3D transformation & viewing [6]

7.1 3D transformations: translation, rotation, scaling & other transformations;

7.2 Three dimensional viewing, Parallel and Perspective projections,

7.3 View Volumes and General Projection Transformations.

7.4 3 D clipping

8 Hidden surfaces Elimination [4]

8.1 Depth comparison, A-buffer algorithm, Back face detection; Depth -Buffer

8.2 Scan-line Method - BSP tree method, the Painter's algorithm, Area-subdivision algorithm;

Text Books:

1. Hearn, Baker – " Computer Graphics (C version 2nd Ed.)" – Pearson education
2. F. S. Hill, Stephen Kelly, Computer Graphics using OpenGL, PHI Learning
3. David F. Rogers - Procedural Elements of Computer Graphics, Tata McGRAw Hill

Reference Books:

4. Foley, Vandam, Feiner, Hughes – "Computer Graphics principles (2nd Ed.) – Pearson Education.
5. W. M. Newman, R. F. Sproull – "Principles of Interactive computer Graphics" – TMH.
6. D. F. Rogers, J. A. Adams – " Mathematical Elements for Computer Graphics (2nd Ed.)" – TMH
7. Z. Xiang, R. Plastock – " Schaum's outlines Computer Graphics (2nd Ed.)" – TMH