COURSE PLAN – COMPUTER GRPHICS LAB

A. OUTLINE OF PRACTICALS

Experiment 1: Introduction to OpenGL
Experiment 2: Drawing line using DDA, Bresenham's algorithm
Experiment 3: Drawing Circle and Ellipse using Mid-point algorithm
Experiment 4: Filling the objects using flood fill, boundary fill and scan line fill algorithm.
Experiment 5&6: Performing Clipping operation and polygon using Cohen Sutherland and Sutherland Hodgeman algorithms respectively.
Experiment 7&8: Performing 2D & 3D TRANSFORMATIONS on objects.
Experiment 9: Drawing Bezier curves.
Experiment 10: Animation & Event Handling using Mouse and Keyboard
Experiment 11&12: Creating 3D Shapes like Cube, Sphere and others.

B. PEDAGOGY

1. **Students need to maintain a practical file which will contain all the executed experiments; the file should contain all the output of all experiments; students will be evaluated based on that file.**
2. **Students should carry a mini lab copy which contains discussion of teachers note or algorithms of the experiments which will be executed.**

C. COURSE COMPLETION PLAN

Total Lab sessions	12

One Session =120 minutes

D. DETAILED LAB EXERCISE PLAN

Experiment 1: Introduction to OpenGL: [Virtual Lab Environment Setup]

- **What is OpenGL?**
- **What is GLU/GLUT?**
- **What is OpenGL Architecture?**
- **Setting up the environment.**
- **First OpenGL Program: This initializes a window of Green color.**

# Discuss all the steps & functions in the program.

Experiment 2: Drawing a line [Usage of Open GL on Linux Environment for Virtual Environment]

- **Draw a line using equation of line Y=m*X+C.**
- **Draw a line using DDA algorithm for slope m<1 and m>1.**
- **Draw a line using Bresenham algorithm for slope m<1 and m>1.**

# Take the input from user for all the three scenarios i.e. value of (x1, y1) and (x2, y2).

Experiment 3: Drawing a Circle and an Ellipse [Done on OpenGL Environment]

- **Draw the circle with the help of polar equations**
- **Draw the circle with the help of mid-point method.**
- **Draw the Ellipse with the mid-point method.**

# Take the value of radius, major axis and minor axis as input from the user.

Experiment 4: Filling –Area [Small Project will be given for demonstration]

- **WAP to fill the polygon using scan lines.**
- **WAP to fill a region using boundary fill algorithm using 4 or 8 connected approaches.**
- **WAP to fill a region using flood fill algorithm using 4 or 8 connected approaches.**

# Take the value of seed point, intensity of new color as input from user.

Experiment 5 & 6: Viewing and Clipping [Geographical Animation for demonstration]

- **Write an interactive program for line clipping using Cohen Sutherland line clipping algorithm.**
- **Write an interactive program for line clipping using Liang-Barsky line clipping algorithm.**
- **Write an interactive program for polygon clipping using Sutherland – Hodgeman polygon clipping algorithm.**

# Take the window coordinates as input from the user, also take polygon coordinates as input.

Experiment 7 & 8: Basic Two3 & Three Dimensional Transformations

- **Write an interactive program for following basic transformation.**
- **Translation**
- **Rotation**
- **Scaling**
- **Reflection about axis.**
- **Reflection about a line Y=mX+c and aX+bY+c=0.**
- **Shear about an edge and about a vertex.**

# Perform all the experiment for 3-D transformation.
# Take the following values as input from user: Theta (angle of rotation),translation factor, scaling factor and other values. Make necessary assumptions.

Experiment 9: Drawing Bezier curves. [Virtual GLUT based demonstration]

- **Write a program to draw a cubic spline.**
- **WAP to draw a Bezier curve.**

# Take necessary values as input from the user like degree of the Bezier curve.

Experiment 10: Animation & Event Handling using Mouse and Keyboard

- **WAP to implement following scenarios**
  - **Mouse Handling**
  - **Mouse Motion Handling**
  - **Keyboard Handling**
  - **Animation Using Mouse**

# Take necessary values as input from the user like time, how long you want animation to run.

Experiment 11&12: Creating 3D Shapes like Cube, Sphere and others.

- **WAP to create various 3D objects:**
  - **CUBE**
  - **SPHERE**
  - **CONE**
  - **TEAPOT.**

#Make necessary assumption for creating the 3-D objects, you can use inbuilt functions to simplify the coding, lightning and shading effect should also be there.

Suggestive reads:

1. OpenGL: Programming Guide, the Official Guide to Learning OpenGL.

Authors: Dave Shreiner, John Kessenich, Bill Licea-Kane,The Khronos OpenGL ARB Working Group.

2. OpenGL Programming Guide Paperback – 2008

Author by Mason Woo (Author), Dave Shreiner (Author)