COMPUTER GRAPHICS AND VISUALIZATION LAB-1

CB.EN.U4CSE22328

**1.Open GL Code for DAA**

**Code:**

#include<stdlib.h>

#include<stdio.h>

#include <GL/gl.h>

#include <GL/glut.h>

float x1, x2, y1, y2;

void display(void) {

  float dy, dx, step, x, y, k, Xin, Yin;

  dx = x2 - x1;

  dy = y2 - y1;

  if (abs(dx) > abs(dy)) {

    step = abs(dx);

  } else

    step = abs(dy);

  Xin = dx / step;

  Yin = dy / step;

  x = x1;

  y = y1;

  glBegin(GL\_POINTS);

  glVertex2i(x, y);

  glEnd();

  for (k = 1; k <= step; k++) {

    x = x + Xin;

    y = y + Yin;

    glBegin(GL\_POINTS);

    glVertex2i(x, y);

    glEnd();

  }

  glFlush();

}

void myInit (void) {

  glClearColor(0.0, 0.0, 0.0, 0.0);

  glMatrixMode(GL\_PROJECTION);

  glLoadIdentity();

  gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

int main(int argc, char \*\* argv) {

  printf("Value of x1 : ");

  scanf("%f", & x1);

  printf("Value of y1 : ");

  scanf("%f", & y1);

  printf("Value of x2 : ");

  scanf("%f", & x2);

  printf("Value of y2 : ");

  scanf("%f", & y2);

  glutInit( & argc, argv);

  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

  glutInitWindowSize(640, 480);

  glutInitWindowPosition(100, 150);

  glutCreateWindow("");

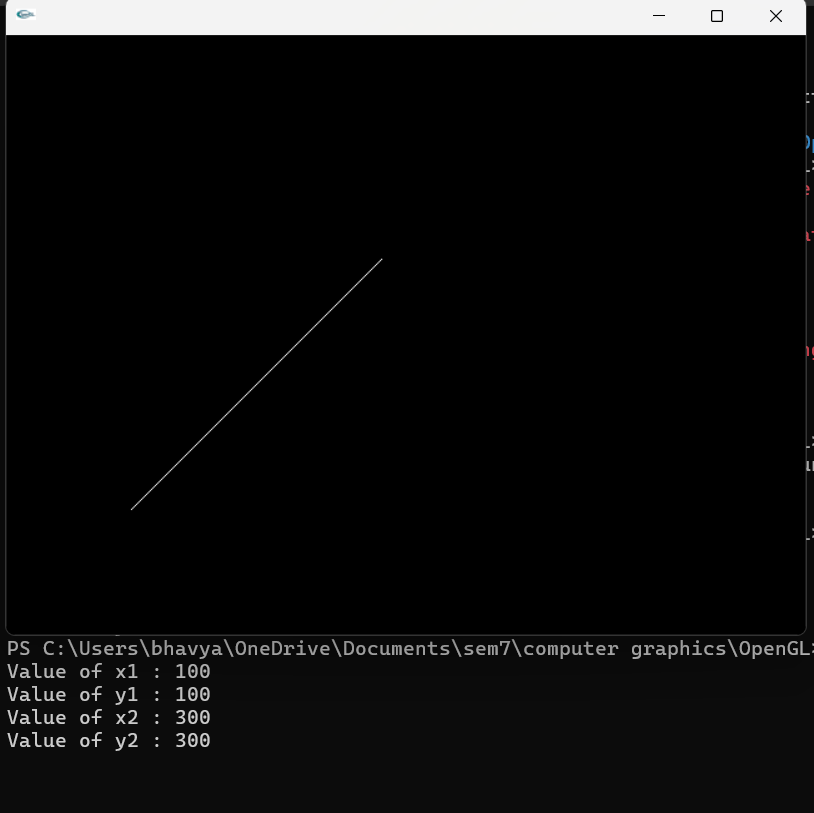
  myInit ();

  glutDisplayFunc(display);

  glutMainLoop();

}

Output:



**2.Boat**

Code:  
#include<stdlib.h>

#include<stdio.h>

#include <GL/gl.h>

#include <GL/glut.h>

void display(void) {

  glClear(GL\_COLOR\_BUFFER\_BIT);

  // Boat base (trapezoid)

  glBegin(GL\_LINE\_LOOP);

    glVertex2i(200, 200);

    glVertex2i(400, 200);

    glVertex2i(350, 150);

    glVertex2i(250, 150);

  glEnd();

  // Sail (triangle)

  glBegin(GL\_LINE\_LOOP);

    glVertex2i(300, 200);

    glVertex2i(300, 300);

    glVertex2i(350, 200);

  glEnd();

  // Mast (vertical line)

  glBegin(GL\_LINES);

    glVertex2i(300, 200);

    glVertex2i(300, 300);

  glEnd();

  glFlush();

}

void myInit(void) {

  glClearColor(0.0, 0.0, 0.0, 0.0); // black background

  glColor3f(1.0, 1.0, 1.0); // white lines

  glMatrixMode(GL\_PROJECTION);

  glLoadIdentity();

  gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

int main(int argc, char \*\*argv) {

  glutInit(&argc, argv);

  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

  glutInitWindowSize(640, 480);

  glutInitWindowPosition(100, 150);

  glutCreateWindow("Simple Boat Outline");

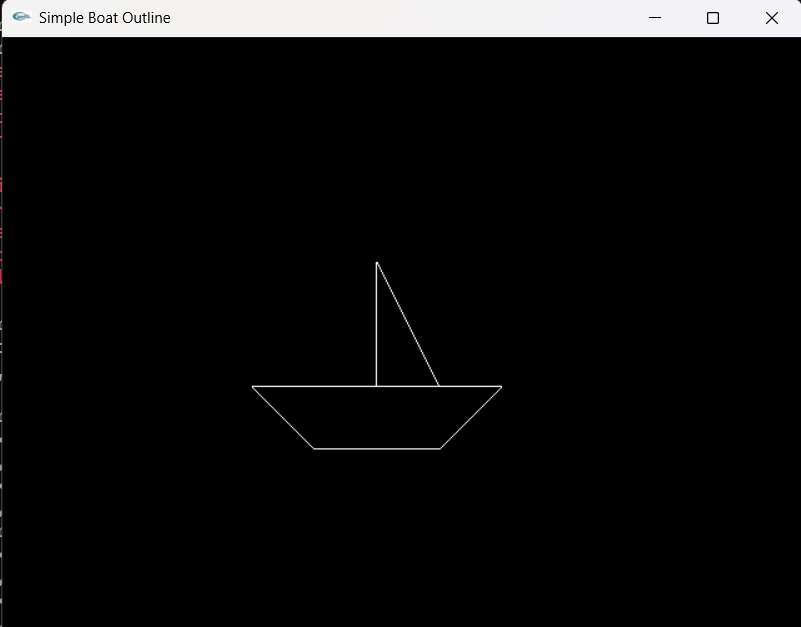
  myInit();

  glutDisplayFunc(display);

  glutMainLoop();

}

**Output:**



**3.House**

**Code:**  
#include<stdlib.h>

#include<stdio.h>

#include <GL/gl.h>

#include <GL/glut.h>

void display(void) {

  glClear(GL\_COLOR\_BUFFER\_BIT);

  // Draw house base (rectangle)

  glBegin(GL\_LINE\_LOOP);

    glVertex2i(200, 200);

    glVertex2i(400, 200);

    glVertex2i(400, 350);

    glVertex2i(200, 350);

  glEnd();

  // Draw roof (triangle)

  glBegin(GL\_LINE\_LOOP);

    glVertex2i(200, 350);

    glVertex2i(400, 350);

    glVertex2i(300, 450);

  glEnd();

  // Draw door (rectangle)

  glBegin(GL\_LINE\_LOOP);

    glVertex2i(270, 200);

    glVertex2i(330, 200);

    glVertex2i(330, 280);

    glVertex2i(270, 280);

  glEnd();

  glFlush();

}

void myInit(void) {

  glClearColor(0.0, 0.0, 0.0, 0.0); // black background

  glColor3f(1.0, 1.0, 1.0); // default drawing color (white)

  glMatrixMode(GL\_PROJECTION);

  glLoadIdentity();

  gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

int main(int argc, char \*\*argv) {

  glutInit(&argc, argv);

  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

  glutInitWindowSize(640, 480);

  glutInitWindowPosition(100, 150);

  glutCreateWindow("Simple House Outline");

  myInit();

  glutDisplayFunc(display);

  glutMainLoop();

}

Output:  
