Experiment – 5

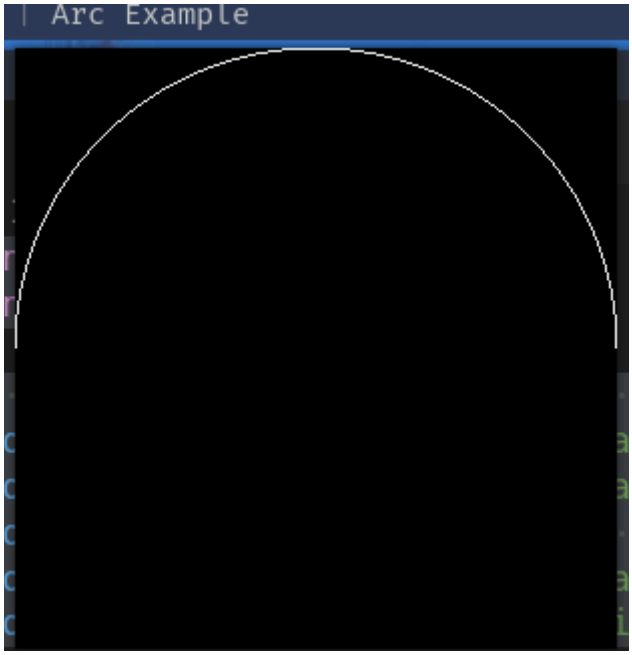
Aim:

Write a program to draw arc, line, circle and rectangle using in built functions.

Arc implementaion:

```
#include <GL/glut.h>
#include <math.h>

// Define the parameters for the arc
float cx = 0.0;     // X-coordinate of center of arc
float cy = 0.0;     // Y-coordinate of center of arc
float r = 100.0;    // Radius of arc
float start_angle = 45.0;  // Starting angle of arc (in degrees)
float end_angle = 135.0;  // Ending angle of arc (in degrees)

// Define the display function
void display() {
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1.0, 0.0, 0.0);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   glTranslatef(cx, cy, 0.0);
   glRotatef(-90.0, 0.0, 0.0, 1.0);  // Rotate the arc to start from x-axis
   gluPartialDisk(gluNewQuadric(), r, r, 50, 1, start_angle, end_angle - start_angle);
   glutSwapBuffers();
}

// Define the reshape function
void reshape(int w, int h) {
   glViewport(0, 0, w, h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(-w/2, w/2, -h/2, h/2);
```

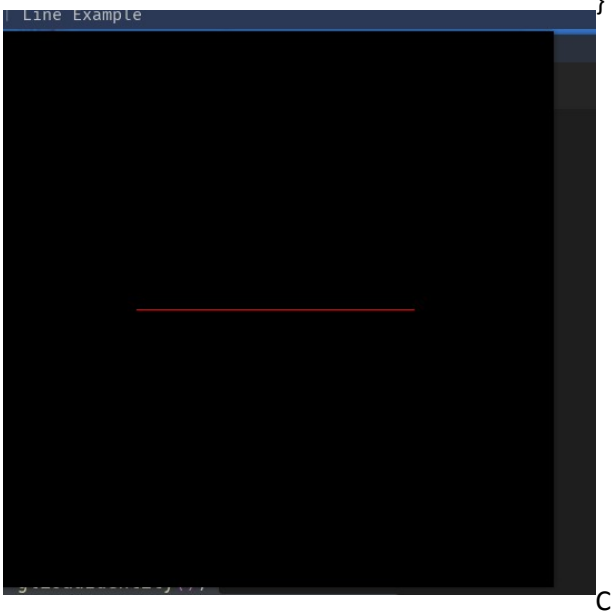
```
}

// Main function
int main(int argc, char **argv) {
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
   glutInitWindowSize(500, 500);
   glutCreateWindow("Arc Example");
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutMainLoop();
   return 0;
}
```



Line implementation:

```
#include <GL/glut.h>

// Define the display function
void display() {
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1.0, 0.0, 0.0);
   glBegin(GL_LINES);
      glVertex2f(-0.5, 0.0);
      glVertex2f(0.5, 0.0);
```

```
    glEnd();
    glutSwapBuffers();
}

// Define the reshape function
void reshape(int w, int h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-1.0, 1.0, -1.0, 1.0);
}

// Main function
int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Line Example");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return 0;
}
```

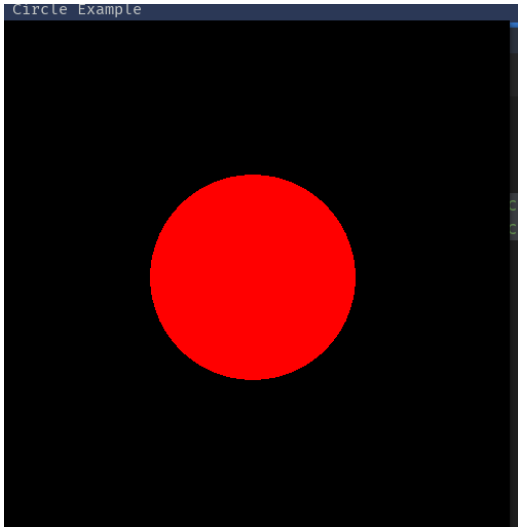


C

Circle Implementation:

```
#include <GL/glut.h>

// Define the parameters for the circle
float cx = 0.0;     // X-coordinate of center of circle
float cy = 0.0;     // Y-coordinate of center of circle
float r = 100.0;    // Radius of circle

// Define the display function
void display() {
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1.0, 0.0, 0.0);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   glTranslatef(cx, cy, 0.0);
   gluDisk(gluNewQuadric(), 0.0, r, 50, 1);
   glutSwapBuffers();
}

// Define the reshape function
void reshape(int w, int h) {
   glViewport(0, 0, w, h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(-w/2, w/2, -h/2, h/2);
}

// Main function
int main(int argc, char **argv) {
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
   glutInitWindowSize(500, 500);
   glutCreateWindow("Circle Example");
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutMainLoop();
```

```
    return 0;
}
```



Rectangle implementation:

```
#include <GL/glut.h>

// Define the parameters for the rectangle
float x1 = -50.0;   // X-coordinate of top-left corner of rectangle
float y1 = 50.0;    // Y-coordinate of top-left corner of rectangle
float x2 = 50.0;    // X-coordinate of bottom-right corner of rectangle
float y2 = -50.0;   // Y-coordinate of bottom-right corner of rectangle

// Define the display function
void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef(0.0, 0.0, 0.0);
    glRectf(x1, y1, x2, y2);
    glutSwapBuffers();
}

// Define the reshape function
void reshape(int w, int h) {
    glViewport(0, 0, w, h);
```

```
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-w/2, w/2, -h/2, h/2);
}

// Main function
int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(250, 250);
    glutCreateWindow("Rectangle Example");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return 0;
}
```