Experiment 2:

Aim: Develop a program to demonstrate basic geometric operations on the 2D object

Program:

```
#include <GL/glut.h>
#include <iostream>
// Global variables
int width = 800;
int height = 600;
float rectWidth = 100.0f;
float rectHeight = 50.0f;
float rectPositionX = (width - rectWidth) / 2.0f;
float rectPositionY = (height - rectHeight) / 2.0f;
float rotationAngle = 0.0f;
float scaleFactor = 1.0f;
// Function to draw a rectangle
void drawRectangle(float x, float y, float width, float height) {
   glBegin(GL_POLYGON);
   glVertex2f(x, y);
   glVertex2f(x + width, y);
   glVertex2f(x + width, y + height);
   glVertex2f(x, y + height);
   glEnd();
}
// Function to handle display
void display() {
   glClear(GL_COLOR_BUFFER_BIT);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   // Apply transformations
   glTranslatef(rectPositionX, rectPositionY, 0.0f);
   glRotatef(rotationAngle, 0.0f, 0.0f, 1.0f);
   glScalef(scaleFactor, scaleFactor, 1.0f);
   // Draw rectangle
   glColor3f(1.0f, 0.0f, 0.0f); // Red color
   drawRectangle(0.0f, 0.0f, rectWidth, rectHeight);
   glFlush();
}
// Function to handle keyboard events
void keyboard(unsigned char key, int x, int y) {
   switch (key) {
      case 't':
         // Translate the rectangle by 10 units in the x-direction
         rectPositionX += 10.0f;
         break;
      case 'r':
         // Rotate the rectangle by 10 degrees clockwise
         rotationAngle += 10.0f;
         break;
```

```
        case 's':
            // Scale the rectangle by 10% (scaleFactor = 1.1f)
            scaleFactor *= 1.1f;
            break;
        case 'u':
            // Reset transformations (translate back to center, reset rotation and scaling)
            rectPositionX = (width - rectWidth) / 2.0f;
            rectPositionY = (height - rectHeight) / 2.0f;
            rotationAngle = 0.0f;
            scaleFactor = 1.0f;
            break;
        case 27: // Escape key to exit
            exit(0);
            break;
    }
    glutPostRedisplay(); // Trigger a redraw
}
// Function to initialize OpenGL
void initializeOpenGL(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(width, height);
    glutCreateWindow("Geometric Operations in 2D");
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // White background
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, width, 0, height);
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
}
// Main function
int main(int argc, char** argv) {
    initializeOpenGL(argc, argv);
    glutMainLoop();
    return 0;
}
```