Task 1

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



void init(void)
{
glClearColor(1.0, 1.0, 1.0, 0.0); // Set display window colour to white



glMatrixMode(GL_PROJECTION); // Set projection parameters
gluOrtho2D(0.0, 35.0, 0.0, 30.0);
}



void drawShapes(void)
{
glClear(GL_COLOR_BUFFER_BIT); // Clear display window



//Set colour to black
glColor3f(0.0, 0.0, 0.0);
```

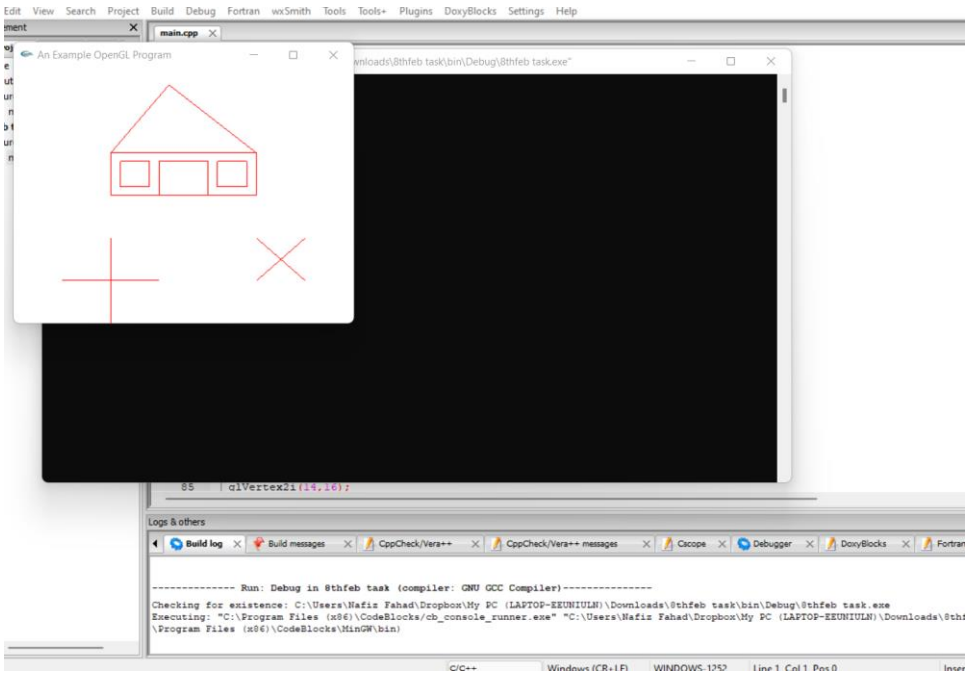
```
//Adjust the point size
glPointSize(10.0);



//Set colour to red
glColor3f(1.0, 0.0, 0.0);



// Draw a + sign outlined triangle
glBegin(GL_LINES);



// Draw a + sign outlined triangle


//for +sign
glVertex2i(5, 5);
glVertex2i(15, 5);

glVertex2i(10, 0);
glVertex2i(10,10);

//for multiply sign
```

```
glVertex2i(25,5);
glVertex2i(30,10);

glVertex2i(25,10);
glVertex2i(30,5);

//for draw a house
glVertex2i(10,15);
glVertex2i(25,15);

glVertex2i(25,15);
glVertex2i(25,20);

glVertex2i(25,20);
glVertex2i(10,20);

glVertex2i(10,20);
glVertex2i(10,15);

//draw door
glVertex2i(15,15);
glVertex2i(15,19);

glVertex2i(15,19);
glVertex2i(20,19);
```

```
glVertex2i(20,19);
glVertex2i(20,15);

//draw 1st window
glVertex2i(11,16);
glVertex2i(14,16);

glVertex2i(14,16);
glVertex2i(14,19);

glVertex2i(14,19);
glVertex2i(11,19);

glVertex2i(11,19);
glVertex2i(11,16);

//draw 2nd window
glVertex2i(21,16);
glVertex2i(24,16);

glVertex2i(24,16);
glVertex2i(24,19);

glVertex2i(24,19);
glVertex2i(21,19);
```

```
glVertex2i(21,19);
glVertex2i(21,16);

//draw roof
glVertex2i(16,28);
glVertex2i(10,20);

glVertex2i(16,28);
glVertex2i(25,20);



glEnd();




glFlush(); // Process all OpenGL routines
}



int main(int argc, char* argv[])
{
glutInit(&argc, argv); // Initalise GLUT
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); // Set display mode
```

```
glutInitWindowPosition(50, 100); // Set window position
glutInitWindowSize(400, 300); // Set window size
glutCreateWindow("An Example OpenGL Program"); // Create display window



init(); // Execute initialisation procedure
glutDisplayFunc(drawShapes); // Send graphics to display window
glutMainLoop(); // Display everything and wait



return 0;
}
```



Task 3

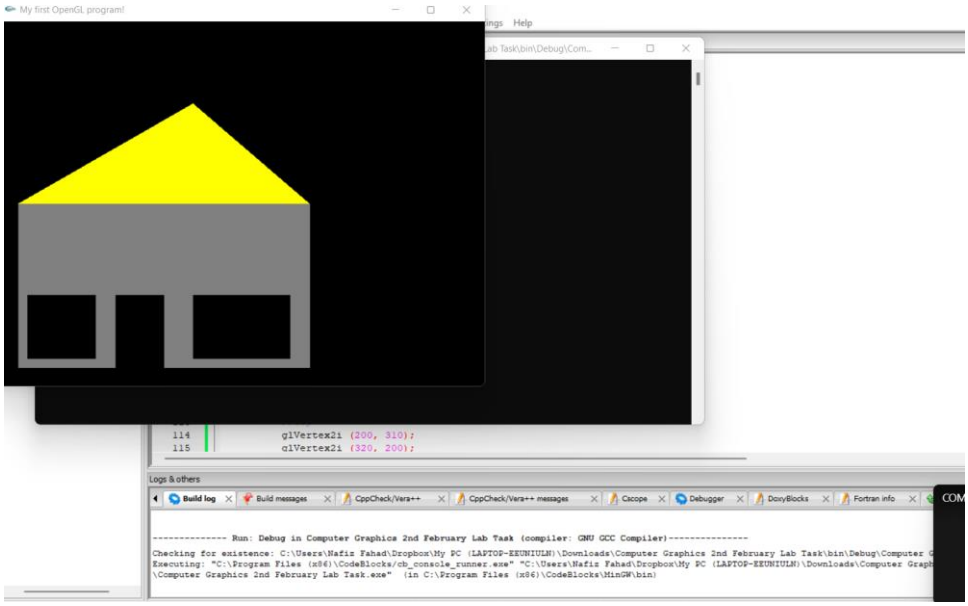
```
#include <GL/gl.h>
#include <GL/glut.h>
void myInit (void)

{

    glClearColor(0.0,0.0,0.0,0.0); // sets background color to white

    // sets a point to be 4x4 pixels

    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 500.0, 0.0, 400.0); // the display area in world coordinates.

}


void myDisplay(void)

{

    glClear(GL_COLOR_BUFFER_BIT); // clears the screen
   glColor3f(0.5f,0.5f,0.5f) ;// setsthe drawing colour
```

```
    glPointSize(4.0);

    glBegin(GL_POLYGON);
//bottom

        glVertex2i (20, 20);
        glVertex2i (320, 20);

        //top
        glVertex2i (20, 200);
        glVertex2i (320, 200);
       //left
        glVertex2i (20, 200);
        glVertex2i (20, 20);
       //right
        glVertex2i (320, 200);
        glVertex2i (320, 20);

        //glVertex2i (10, -50);

   glEnd();


//Left Window

glColor3f(0.0f,0.0f,0.0f) ;
glPointSize(4.0);
glRectf(30,30,100,100);

//Right Window
```

```
glColor3f(0.0f,0.0f,0.0f) ;
glBegin(GL_POLYGON);
//bottom

      glVertex2i (200,30);
       glVertex2i (200, 100);

//top
       glVertex2i (200, 100);
       glVertex2i (300, 100);

//left
       glVertex2i (300, 100);
       glVertex2i (300, 30);
//right
      glVertex2i (300, 30);
       glVertex2i (200, 30);

       //glVertex2i (10, -50);

  glEnd();


//door
glColor3f(0.0f,0.0f,0.0f) ;
glBegin(GL_POLYGON);
//left

      glVertex2i (120,20);
       glVertex2i (120, 100);
```

```
//top
        glVertex2i (120, 100);
        glVertex2i (170, 100);

//right
        glVertex2i (170, 100);
        glVertex2i (170, 20);

//bottom
       glVertex2i (120, 20);
        glVertex2i (170, 20);

        //glVertex2i (10, -50);

   glEnd();

   //Triangle

   glColor3f(1.0f,1.0f,0.0f) ;
   glBegin(GL_POLYGON);
//bottom

        glVertex2i (20, 200);
        glVertex2i (320, 200);

        //top
        glVertex2i (200, 310);
        glVertex2i (320, 200);

       //left
        glVertex2i (20, 200);
```

```
        glVertex2i (200, 310);


        //glVertex2i (10, -50);

    glEnd();




    glFlush(); // sends all output to display;

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

    glutInit (&argc, argv); // to initialize the toolkit;

    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); // sets the display mode

    glutInitWindowSize (640, 480); // sets the window size

    glutInitWindowPosition (10, 10); // sets the starting position for the window

    glutCreateWindow ("My first OpenGL program!"); // creates the window and sets the title

    glutDisplayFunc (myDisplay);
    myInit(); // additional initializations as necessary
    glutMainLoop(); // go into a loop until event occurs
    return 0;
}
```



Task 2

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



void init(void)
{
glClearColor(1.0, 1.0, 1.0, 0.0); // Set display window colour to white



glMatrixMode(GL_PROJECTION); // Set projection parameters
gluOrtho2D(0.0, 35.0, 0.0, 30.0);
}
```

```
void drawShapes(void)
{
glClear(GL_COLOR_BUFFER_BIT); // Clear display window



//Set colour to black
glColor3f(0.0, 0.0, 0.0);
//Adjust the point size
glPointSize(10.0);



//Set colour to red
glColor3f(1.0, 0.0, 0.0);


glBegin(GL_TRIANGLES); //Begin triangle coordinates



//Triangle
glColor3f (0.0, 0.5, 1.0);
```

```
glVertex3f(10, 15, 0);
glVertex3f(10, 25, 0);
glVertex3f(25, 15, 0);

glColor3f (1.0, 0.0, 0.0);
glVertex3f(10, 25, 0);
glVertex3f(25, 25, 0);
glVertex3f(25, 15, 0);

glColor3f (0.0, 0.0, 0.0);
glVertex3f(10, 25, 0);
glVertex3f(25, 25, 0);
glVertex3f(18, 29, 0);

glEnd();//End triangle coordinates

glBegin(GL_QUADS);

glColor3f(0.3, 0.4, 1);
```

```
glVertex2f(5, 12);
glVertex2f(30, 12);
glVertex2f(25, 2);
glVertex2f(10, 2);



glEnd();



glFlush ();
}
int main(int argc, char* argv[])
{
glutInit(&argc, argv); // Initalise GLUT
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); // Set display mode



glutInitWindowPosition(50, 100); // Set window position
glutInitWindowSize(400, 300); // Set window size
glutCreateWindow("An Example OpenGL Program"); // Create display
window



init(); // Execute initialisation procedure
glutDisplayFunc(drawShapes); // Send graphics to display window
```

glutMainLoop(); // Display everything and wait

return 0;

}