

Makeup Lab class Tasks

Name: Md. Abdul Muneem Adnan

ID: 20-44213-3

Question- 1

Draw a bench.

Graph Plot (Picture)-

Code-

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

#include <windows.h>

void drawBench() {

    glPushMatrix();

    glTranslatef(-1.5, -0.5, -1.0);

    glScalef(0.1, 1.0, 0.1);

    glutSolidCube(1.0);

    glPopMatrix();

    glPushMatrix();

    glTranslatef(1.5, -0.5, -1.0);
```

```
glScalef(0.1, 1.0, 0.1);
```

```
glutSolidCube(1.0);
```

```
glPopMatrix();
```

```
glPushMatrix();
```

```
glTranslatef(0.0, 0.0, -1.0);
```

```
glScalef(3.0, 0.1, 1.0);
```

```
glutSolidCube(1.0);
```

```
glPopMatrix();
```

```
glPushMatrix();
```

```
glTranslatef(0.0, 0.5, -1.0);
```

```
glScalef(3.0, 0.1, 0.1);
```

```
glutSolidCube(1.0);
```

```
glPopMatrix();
```

```
glPushMatrix();
```

```
glTranslatef(0.0, 0.75, -1.0);
```

```
glScalef(3.0, 0.05, 0.05);
```

```
glutSolidCube(1.0);
```

```
glPopMatrix();
```

```
glPushMatrix();
```

```
glTranslatef(0.0, 0.25, -0.6);
```

```
glScalef(3.0, 0.05, 0.05);
```

```
    glutSolidCube(1.0);

    glPopMatrix();
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glLoadIdentity();

    glTranslatef(0.0, 0.0, -5.0);

    glColor3f(0.53, 0.81, 0.92);
    drawBench();

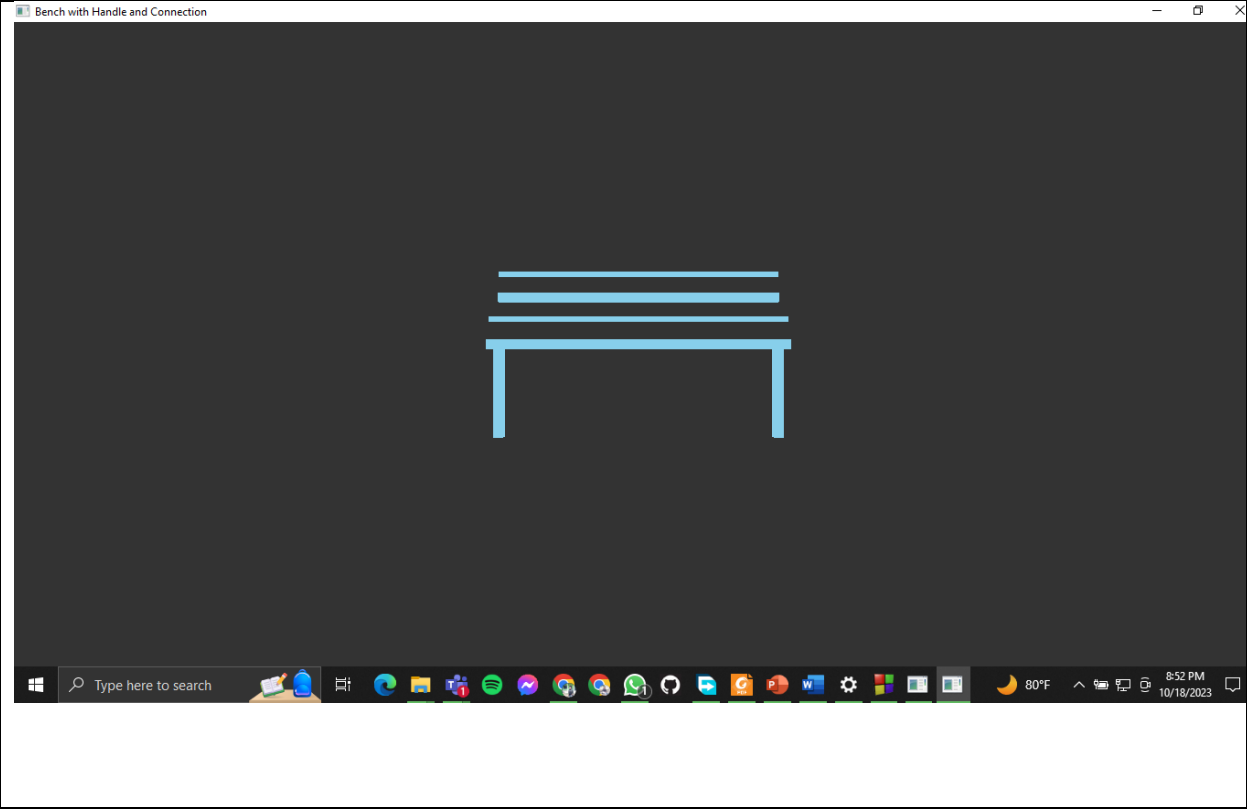
    glutSwapBuffers();
}

void reshape(int w, int h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, (GLfloat)w / (GLfloat)h, 1.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
}

void init() {
```

```
    glClearColor(0.2, 0.2, 0.2, 1.0);  
}  
  
int main(int argc, char** argv) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);  
    glutInitWindowSize(800, 600);  
    glutCreateWindow("Bench with Handle and Connection");  
    glutDisplayFunc(display);  
    glutReshapeFunc(reshape);  
  
    init();  
  
    glEnable(GL_DEPTH_TEST);  
  
    glutMainLoop();  
  
    return 0;  
}
```

Output Screenshot (Full Screen)-



Question- 2



1-Denmark.png



1-Tonga.png



1-Vietnam.png



1-Somalia.jpg



1-Switzerland.pn

g

Graph Plot (Picture)-

Code-

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

#include <cmath>

void drawDenmarkFlag() {

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(0.678, 0.078, 0.078);

    glBegin(GL_POLYGON);
```

```
glVertex2f(-1.0, 1.0);  
glVertex2f(1.0, 1.0);  
glVertex2f(1.0, -1.0);  
glVertex2f(-1.0, -1.0);  
glEnd();
```

```
glColor3f(1.0, 1.0, 1.0);  
glBegin(GL_POLYGON);  
glVertex2f(-0.1, 1.0);  
glVertex2f(0.1, 1.0);  
glVertex2f(0.1, -1.0);  
glVertex2f(-0.1, -1.0);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glVertex2f(-1.0, 0.1);  
glVertex2f(1.0, 0.1);  
glVertex2f(1.0, -0.1);  
glVertex2f(-1.0, -0.1);  
glEnd();
```

```
}
```

```
void drawSwitzerlandFlag() {  
    glClear(GL_COLOR_BUFFER_BIT);
```

```
glColor3f(1.0, 0.0, 0.0);  
glBegin(GL_POLYGON);  
glVertex2f(-1.0, 1.0);  
glVertex2f(1.0, 1.0);  
glVertex2f(1.0, -1.0);  
glVertex2f(-1.0, -1.0);  
glEnd();
```

```
glColor3f(1.0, 1.0, 1.0);  
glBegin(GL_POLYGON);  
glVertex2f(-0.05, 0.5);  
glVertex2f(0.05, 0.5);  
glVertex2f(0.05, -0.5);  
glVertex2f(-0.05, -0.5);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glVertex2f(-0.5, 0.05);  
glVertex2f(0.5, 0.05);  
glVertex2f(0.5, -0.05);  
glVertex2f(-0.5, -0.05);  
glEnd();
```

```
glColor3f(1.0, 1.0, 1.0);  
glBegin(GL_POLYGON);
```



```
    glVertex2f(-0.05, 0.4);

    glVertex2f(0.05, 0.4);

    glVertex2f(0.05, -0.4);

    glVertex2f(-0.05, -0.4);

    glEnd();


    glBegin(GL_POLYGON);

    glVertex2f(-0.4, 0.05);

    glVertex2f(0.4, 0.05);

    glVertex2f(0.4, -0.05);

    glVertex2f(-0.4, -0.05);

    glEnd();
}


void drawVietnamFlag() {

    glColor3f(1.0, 0.0, 0.0);

    glBegin(GL_QUADS);

    glVertex2f(-1.0, 1.0);

    glVertex2f(1.0, 1.0);

    glVertex2f(1.0, -1.0);

    glVertex2f(-1.0, -1.0);

    glEnd();


    glColor3f(1.0, 1.0, 0.0);

    glBegin(GL_POLYGON);
```

```
for (int i = 0; i < 5; i++) {  
  
    float angle = i * (360.0 / 5);  
  
    float x = 0.0 + 0.3 * cos(angle * 3.14159265 / 180);  
  
    float y = 0.0 + 0.3 * sin(angle * 3.14159265 / 180);  
  
    glVertex2f(x, y);  
  
}  
  
glEnd();  
}
```

```
void drawSomaliaFlag() {  
  
    glColor3f(0.0, 0.447, 0.741);  
  
    glBegin(GL_QUADS);  
  
    glVertex2f(-1.0, 1.0);  
  
    glVertex2f(1.0, 1.0);  
  
    glVertex2f(1.0, -1.0);  
  
    glVertex2f(-1.0, -1.0);  
  
    glEnd();  
  
  
    glColor3f(1.0, 1.0, 1.0);  
  
    glBegin(GL_POLYGON);  
  
    for (int i = 0; i < 5; i++) {  
  
        float angle = i * (360.0 / 5);  
  
        float x = 0.0 + 0.3 * cos(angle * 3.14159265 / 180);  
  
        float y = 0.0 + 0.3 * sin(angle * 3.14159265 / 180);  
  
        glVertex2f(x, y);  

```

```
}  
  
glEnd();  
  
}  
  
void drawTongaFlag() {  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    glColor3f(1.0, 0.0, 0.0);  
    glBegin(GL_QUADS);  
    glVertex2f(-1.0, -1.0);  
    glVertex2f(1.0, -1.0);  
    glVertex2f(1.0, 1.0);  
    glVertex2f(-1.0, 1.0);  
    glEnd();  
  
    glColor3f(1.0, 1.0, 1.0);  
    glBegin(GL_QUADS);  
    glVertex2f(-1.0, 0.5);  
    glVertex2f(0.0, 0.5);  
    glVertex2f(0.0, 1.0);  
    glVertex2f(-1.0, 1.0);  
    glEnd();  
  
    glColor3f(1.0, 0.0, 0.0);  
    glBegin(GL_QUADS);
```

```
    glVertex2f(-0.9, 0.5);

    glVertex2f(-0.8, 0.5);

    glVertex2f(-0.8, 0.8);

    glVertex2f(-0.9, 0.8);

    glEnd();


    glBegin(GL_QUADS);

    glVertex2f(-1.0, 0.6);

    glVertex2f(-0.7, 0.6);

    glVertex2f(-0.7, 0.7);

    glVertex2f(-1.0, 0.7);

    glEnd();
}


void reshape(int width, int height) {

    glViewport(0, 0, width, height);

    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();

    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);

    glMatrixMode(GL_MODELVIEW);

}


void displayDenmark() {

    glClear(GL_COLOR_BUFFER_BIT);
```

```
drawDenmarkFlag();

glFlush();

}

void displaySwitzerland() {

    glClear(GL_COLOR_BUFFER_BIT);

    drawSwitzerlandFlag();

    glFlush();

}

void displayVietnam() {

    glClear(GL_COLOR_BUFFER_BIT);

    drawVietnamFlag();

    glFlush();

}

void displaySomalia() {

    glClear(GL_COLOR_BUFFER_BIT);

    drawSomaliaFlag();

    glFlush();

}

void displayTonga() {

    glClear(GL_COLOR_BUFFER_BIT);

    drawTongaFlag();
```

```
glFlush();  
  
}  
  
int main(int argc, char** argv) {  
  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  
  
    glutInitWindowSize(300, 300);  
    glutCreateWindow("Denmark Flag");  
    glutDisplayFunc(displayDenmark);  
  
    glutInitWindowSize(300, 300);  
    glutCreateWindow("Switzerland Flag");  
    glutDisplayFunc(displaySwitzerland);  
  
    glutInitWindowSize(300, 300);  
    glutCreateWindow("Vietnam Flag");  
    glutDisplayFunc(displayVietnam);  
  
    glutInitWindowSize(300, 300);  
    glutCreateWindow("Somalia Flag");  
    glutDisplayFunc(displaySomalia);  
  
    glutInitWindowSize(300, 300);  
    glutCreateWindow("Tonga Flag");
```

```
glutDisplayFunc(displayTonga);
```

```
glutMainLoop();
```

```
return 0;
```

```
}
```

Output Screenshot (Full Screen)-

