**Lab Taks-6**

|  |
| --- |
| **Question-**  Develop an animation that will change the background color of the window after 20ms. Use at least two different colors. |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  void display(int val);  void blue()  {  glClearColor(0.0f, 0.0f, 1.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glBegin(GL\_LINES);  glColor3ub(0, 0, 255);  glEnd();  glutTimerFunc(20, display, 0);  glutPostRedisplay();  glFlush();  }  void disBlue(int val)  {  glutDisplayFunc(blue);  }  void red()  {  glClearColor(1.0f, 0.0f, 0.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glBegin(GL\_LINES);  glColor3ub(255, 0, 0);  glEnd();  glutTimerFunc(20, disBlue, 0);  glutPostRedisplay();  glFlush();  }  void disRed(int val)  {  glutDisplayFunc(red);  }  void display(int val)  {  glClearColor(0.7f, 0.7f, 0.65f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glutTimerFunc(20, disRed, 0);  glutPostRedisplay();  glFlush();  }  void dis()  {  glutTimerFunc(20, display, 0);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(600, 600);  glutCreateWindow("lab task 6");  glutDisplayFunc(dis);  gluOrtho2D(-250,250,-250,250);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop an animation that will call four objects separately, each after 20 ms. |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  void red(int val)  {  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glLineWidth(5);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glutPostRedisplay();  glFlush();  }  void blue(int val)  {  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 255);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glutPostRedisplay();  glFlush();  }  void green(int val)  {  glBegin(GL\_POLYGON);  glColor3ub(0, 255, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glutPostRedisplay();  glFlush();  }  void yellow(int val)  {  glBegin(GL\_POLYGON);  glColor3ub(255, 255, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glutPostRedisplay();  glFlush();  }  void startDisplay()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.7f, 0.7f, 0.65f, 1.0f);  glutTimerFunc(20, red, 0);  glutTimerFunc(40, blue, 0);  glutTimerFunc(60, green, 0);  glutTimerFunc(80, yellow, 0);  glutPostRedisplay();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(800, 800);  glutCreateWindow("lab task 6");  glutDisplayFunc(startDisplay);  gluOrtho2D(-200,200,-200,200);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |
| **Question-**  Develop a code that will have four different objects (keep it simple). The objects will move to the left, right, up and down in a loop. |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  float \_move1 = 3.0f;  void update1(int value) {  \_move1+=5;  if(\_move1 > 300)  {  \_move1=-300;  }  glutPostRedisplay();  glutTimerFunc(20, update1, 0);  }  float \_move2 = 3.0f;  void update2(int value) {  \_move2-=5;  if(\_move2 < -300)  {  \_move2=+300;  }  glutPostRedisplay();  glutTimerFunc(20, update2, 0);  }  float \_move3 = 3.0f;  void update3(int value) {  \_move3-=5;  if(\_move3 < -300)  {  \_move3=+300;  }  glutPostRedisplay();  glutTimerFunc(20, update3, 0);  }  float \_move4 = 3.0f;  void update4(int value) {  \_move4+=5;  if(\_move4 > +300)  {  \_move4=-300;  }  glutPostRedisplay();  glutTimerFunc(20, update4, 0);  }  void red()  {  glPushMatrix();  glTranslatef(\_move1,0,0);  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glLineWidth(5);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void blue()  {  glPushMatrix();  glTranslatef(\_move2,0,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 255);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void green()  {  glPushMatrix();  glTranslatef(0,\_move3,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 255, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void yellow()  {  glPushMatrix();  glTranslatef(0,\_move4,0);  glBegin(GL\_POLYGON);  glColor3ub(255, 255, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void startDisplay()  {  glClearColor(0.7f, 0.7f, 0.65f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glMatrixMode(GL\_MODELVIEW);  red();  blue();  green();  yellow();  glutSwapBuffers();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(800, 800);  glutCreateWindow("lab task 6 [22-47226-1]");  glutDisplayFunc(startDisplay);  gluOrtho2D(-200,200,-200,200);  glutTimerFunc(20, update1, 0);  glutTimerFunc(20, update2, 0);  glutTimerFunc(20, update3, 0);  glutTimerFunc(20, update4, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop a code that will have four different objects (keep it simple). Four different keys will be dedicated each objects. The objects will move to the left, right, up and down in a loop as the keys are pressed individually. |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  float \_move1 = 0.0f;  float speed1 = 0.0f;  void update1(int value) {  \_move1+=speed1;  if(\_move1 > 300)  {  \_move1=-300;  }  glutPostRedisplay();  glutTimerFunc(20, update1, 0);  }  float \_move2 = 0.0f;  float speed2 = 0.0f;  void update2(int value) {  \_move2-=speed2;  if(\_move2 < -300)  {  \_move2=+300;  }  glutPostRedisplay();  glutTimerFunc(20, update2, 0);  }  float \_move3 = 0.0f;  float speed3 = 0.0f;  void update3(int value) {  \_move3-=speed3;  if(\_move3 < -300)  {  \_move3=+300;  }  glutPostRedisplay();  glutTimerFunc(20, update3, 0);  }  float \_move4 = 0.0f;  float speed4 = 0.0f;  void update4(int value) {  \_move4+=speed4;  if(\_move4 > +300)  {  \_move4=-300;  }  glutPostRedisplay();  glutTimerFunc(20, update4, 0);  }  void red()  {  glPushMatrix();  glTranslatef(\_move1,0,0);  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glLineWidth(5);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void blue()  {  glPushMatrix();  glTranslatef(\_move2,0,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 255);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void green()  {  glPushMatrix();  glTranslatef(0,\_move3,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 255, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void yellow()  {  glPushMatrix();  glTranslatef(0,\_move4,0);  glBegin(GL\_POLYGON);  glColor3ub(255, 255, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void startDisplay()  {  glClearColor(0.7f, 0.7f, 0.65f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glMatrixMode(GL\_MODELVIEW);  red();  blue();  green();  yellow();  glutSwapBuffers();  glFlush();  }  void handleKeypress(unsigned char key, int x, int y)  {  switch (key)  {  case 'w':  speed4 = 5.0f;  break;  case 'a':  speed2 = 5.1f;  break;  case 's':  speed3 = 5.1f;  break;  case 'd':  speed1 = 5.1f;  break;  case 'p':  speed1 = 0.0f;  speed2 = 0.0f;  speed3 = 0.0f;  speed4 = 0.0f;  break;  glutPostRedisplay();  }  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(800, 800);  glutCreateWindow("lab task 6 [22-47226-1]");  glutDisplayFunc(startDisplay);  gluOrtho2D(-200,200,-200,200);  glutTimerFunc(20, update1, 0);  glutTimerFunc(20, update2, 0);  glutTimerFunc(20, update3, 0);  glutTimerFunc(20, update4, 0);  glutKeyboardFunc(handleKeypress);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop a code that will have four different objects (keep it simple). Two of the objects will move to the right as the right click is made on the mouse and two of the objects will move to the left as the left key is pressed on the mouse. |
| **Code-**  #include <windows.h>  #include<GL/gl.h>  #include <GL/glut.h>  float \_move1 = 0.0f;  float speed1 = 0.0f;  void update1(int value) {  \_move1+=speed1;  if(\_move1 > 300)  {  \_move1=-300;  }  glutPostRedisplay();  glutTimerFunc(20, update1, 0);  }  float \_move2 = 0.0f;  float speed2 = 0.0f;  void update2(int value) {  \_move2-=speed2;  if(\_move2 < -300)  {  \_move2=+300;  }  glutPostRedisplay();  glutTimerFunc(20, update2, 0);  }  void red()  {  glPushMatrix();  glTranslatef(\_move1,0,0);  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glLineWidth(5);  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,80);  glVertex2d(100,80);  glVertex2d(100,0);  glVertex2d(-100,0);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void blue()  {  glPushMatrix();  glTranslatef(\_move2,0,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 255);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-100,0);  glVertex2d(100,0);  glVertex2d(100,-80);  glVertex2d(-100,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void green()  {  glPushMatrix();  glTranslatef(\_move2,0,0);  glBegin(GL\_POLYGON);  glColor3ub(0, 255, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(-150,80);  glVertex2d(-75,80);  glVertex2d(-75,-80);  glVertex2d(-150,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void yellow()  {  glPushMatrix();  glTranslatef(\_move1,0,0);  glBegin(GL\_POLYGON);  glColor3ub(255, 255, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);  glBegin(GL\_POLYGON);  glColor3ub(0, 0, 0);  glVertex2d(75,80);  glVertex2d(150,80);  glVertex2d(150,-80);  glVertex2d(75,-80);  glEnd();  glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL);  glPopMatrix();  glutPostRedisplay();  }  void startDisplay()  {  glClearColor(0.7f, 0.7f, 0.65f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  glMatrixMode(GL\_MODELVIEW);  red();  blue();  green();  yellow();  glutSwapBuffers();  glFlush();  }  void handleMouse(int button, int state, int x, int y)  {  if (button == GLUT\_LEFT\_BUTTON)  {  speed2 += 0.5f;  }  if (button == GLUT\_RIGHT\_BUTTON)  {  speed1 += 0.5f;  }  glutPostRedisplay();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(800, 800);  glutCreateWindow("lab task 6 [22-47226-1]");  glutDisplayFunc(startDisplay);  gluOrtho2D(-200,200,-200,200);  glutTimerFunc(20, update1, 0);  glutTimerFunc(20, update2, 0);  glutMouseFunc(handleMouse);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |