**Lab Taks-6**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* Must include resources for all the section in the table

|  |
| --- |
| **Question-**  Develop an animation that will change the background color of the window after 20ms. Use at least two different colors. |
| **Code-**  #include <GL/gl.h>  #include <GL/glut.h>  int windowWidth = 800;  int windowHeight = 600;  float currentColor[3] = {1.0, 0.0, 0.0};  void changingColor()  {  if (currentColor[0] == 1.0)  {  currentColor[0] = 0.0;  currentColor[2] = 1.0;  }  else  {  currentColor[0] = 1.0;  currentColor[2] = 0.0;  }  glutPostRedisplay();  }  void display()  {  glClearColor(currentColor[0], currentColor[1], currentColor[2], 1.0);  glClear(GL\_COLOR\_BUFFER\_BIT);  glutSwapBuffers();  }  void timer(int value)  {  changingColor();  glutTimerFunc(20, timer, 0); //20ms  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA);  glutInitWindowSize(windowWidth, windowHeight);  glutCreateWindow("Colour Changing Animation");  glutDisplayFunc(display);  glutTimerFunc(0, timer, 0); //start  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-**  Develop an animation that will call four objects separately, each after 20 ms. |
| **Code-**  #include<cstdio>  #include <GL/gl.h>  #include <GL/glut.h>  GLfloat position = 0.0f;  GLfloat position1 = 0.0f;  GLfloat speed = 0.1f;  void dis();  void display();  void update(int value)  {  if(position <-1.5)  {  position = 1.0f;  }  position -= speed;  glutPostRedisplay();  glutTimerFunc(100,update,0);  }  void update1(int value)  {  if(position1 >1.0)  {  position1 = -1.0f;  }  position1 += speed;  glutPostRedisplay();  glutTimerFunc(100,update1,0);  }  void init()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display);  }  void display7()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position1,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(35, 137, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(20,disback,0);  glFlush();  }  void display6(int val)  {  glutDisplayFunc(display7);  }  void display5()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(199, 0, 57 );  glVertex2f(-0.5f, -0.5f);  glVertex2f( 0.5f, -0.5f);  glVertex2f( 0.5f, 0.5f);  glVertex2f(-0.5f, 0.5f);  glEnd();  glPopMatrix();  glutTimerFunc(20,display6,0);  glFlush();  }  void display4(int val)  {  glutDisplayFunc(display5);  }  void display3()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position1,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub( 40, 198, 35);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(20,display4,0);  glFlush();  }  void display2(int val)  {  glutDisplayFunc(display3);  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glLoadIdentity();  glPushMatrix();  glTranslatef(position,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(183, 35, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(20,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(320, 320);  glutInitWindowPosition(50, 50);  glutCreateWindow("Four Object");  glutDisplayFunc(dis);  init();  glutTimerFunc(100, update, 0);  glutTimerFunc(100, update1, 0);  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<cstdio>  #include <GL/gl.h>  #include <GL/glut.h>  GLfloat position = 0.0f;  GLfloat position1 = 0.0f;  GLfloat speed = 0.1f;  void dis();  void display();  void update(int value)  {  if(position <-1.5)  {  position = 1.0f;  }  position -= speed;  glutPostRedisplay();  glutTimerFunc(100,update,0);  }  void update1(int value)  {  if(position1 >1.0)  {  position1 = -1.0f;  }  position1 += speed;  glutPostRedisplay();  glutTimerFunc(100,update1,0);  }  void init()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display);  }  void display7()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position1, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(35, 137, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glFlush();  }  void display6(int val)  {  glutDisplayFunc(display7);  }  void display5()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(199, 0, 57 );  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display6,0);  glFlush();  }  void display4(int val)  {  glutDisplayFunc(display5);  }  void display3()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position1,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub( 40, 198, 35);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display4,0);  glFlush();  }  void display2(int val)  {  glutDisplayFunc(display3);  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glLoadIdentity();  glPushMatrix();  glTranslatef(position,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(183, 35, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(320, 320);  glutInitWindowPosition(50, 50);  glutCreateWindow("Loop Animation");  glutDisplayFunc(dis);  init();  glutTimerFunc(100, update, 0);  glutTimerFunc(100, update1, 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<cstdio>  #include <GL/gl.h>  #include <GL/glut.h>  GLfloat position = 0.0f;  GLfloat position1 = 0.0f;  GLfloat speed = 0.1f;  void dis();  void display();  void update(int value)  {  if(position <-2)  {  position = 1.0f;  }  position -= speed;  glutPostRedisplay();  glutTimerFunc(100,update,0);  }  void update1(int value)  {  if(position1 >1.0)  {  position1 = -1.0f;  }  position1 += speed;  glutPostRedisplay();  glutTimerFunc(100,update1,0);  }  void init()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display);  }  void display7()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position1, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(35, 137, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,disback,0);  glFlush();  }  void display6(int val)  {  glutDisplayFunc(display7);  }  void display5()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(199, 0, 57 );  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display6,0);  glFlush();  }  void display4(int val)  {  glutDisplayFunc(display5);  }  void display3()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position1,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub( 40, 198, 35);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display4,0);  glFlush();  }  void display2(int val)  {  glutDisplayFunc(display3);  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glLoadIdentity();  glPushMatrix();  glTranslatef(position,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(183, 35, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display);  }  void handleKeypress(unsigned char key, int x, int y)  {  switch (key)  {  case 'a':  speed = 0.0f;  break;  case 'b':  speed = 0.1f;  break;  case 'c':  speed -= 0.001f;  break;  case 'd':  speed += 0.001f;  break;  glutPostRedisplay();  }  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(320, 320);  glutInitWindowPosition(50, 50);  glutCreateWindow("Four Key Animation");  glutDisplayFunc(dis);  init();  glutTimerFunc(100, update, 0);  glutTimerFunc(100, update1, 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<cstdio>  #include <GL/gl.h>  #include <GL/glut.h>  GLfloat position = 0.0f;  GLfloat position1 = 0.0f;  GLfloat speed = 0.1f;  void dis();  void display();  void update(int value)  {  if(position <-1.0)  {  position = 1.0f;  }  position -= speed;  glutPostRedisplay();  glutTimerFunc(150,update,0);  }  void update1(int value)  {  if(position1 >1.0)  {  position1 = -1.0f;  }  position1 += speed;  glutPostRedisplay();  glutTimerFunc(150,update1,0);  }  void init()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display);  }  void display7()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position1, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(35, 137, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,disback,0);  glFlush();  }  void display6(int val)  {  glutDisplayFunc(display7);  }  void display5()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(199, 0, 57 );  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display6,0);  glFlush();  }  void display4(int val)  {  glutDisplayFunc(display5);  }  void display3()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position1,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub( 40, 198, 35);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display4,0);  glFlush();  }  void display2(int val)  {  glutDisplayFunc(display3);  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glLoadIdentity();  glPushMatrix();  glTranslatef(position,0.0f, 0.0f);  glBegin(GL\_QUADS);  glColor3ub(183, 35, 198);  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display);  }  void handleMouse(int button, int state, int x, int y)  {  if (button = GLUT\_LEFT\_BUTTON)  {  speed += 0.01f;  }  if (button = GLUT\_RIGHT\_BUTTON)  {  speed -= 0.01f;  }  glutPostRedisplay();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(320, 320);  glutInitWindowPosition(50, 50);  glutCreateWindow("Click Animation");  glutDisplayFunc(dis);  init();  glutTimerFunc(100, update, 0);  glutTimerFunc(100, update1, 0);  glutMouseFunc(handleMouse);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |