**Lab Taks-6**

|  |
| --- |
| **Question-01**  Develop an animation that will change the background color of the window after 20ms. Use at least two different colors. |
| **Code-**  #include <GL/glut.h>  #include<math.h>  using namespace std;  // MD. SHOHANUR RAHMAN SHOHAN  // 22-46013-1  GLfloat position = 0.0f;  GLfloat position1 = 0.0f;  GLfloat speed = 0.1f;  void dis();  void display1();  void update(int value)  {  if(position <-1.5)  position = 1.0f;  position -= speed;  glutPostRedisplay();  glutTimerFunc(1500,update,0);  }  void update1(int value)  {  if(position1 >1.0)  position1 = -1.0f;  position1 += speed;  glutPostRedisplay();  glutTimerFunc(1500,update1,0);  }  void init()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display1);  }  void display3()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 1.0f, 0.0f, 1.0f);  glutTimerFunc(1500,disback,0);  glFlush();  }  void display2(int val)  {  glutDisplayFunc(display3);  }  void display1()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(1.0f, 0.0f, 0.0f, 1.0f);  glutTimerFunc(1500,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display1);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(850, 600);  glutInitWindowPosition(50, 50);  glutCreateWindow("Display change");  glutDisplayFunc(dis);  init();  glutTimerFunc(20, update, 0);  glutTimerFunc(20, update1, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-02**  Develop an animation that will call four objects separately, each after 20 ms. |
| **Code-**  #include <GL/glut.h>  #include<math.h>  using namespace std;  // MD. SHOHANUR RAHMAN SHOHAN  // 22-46013-1  GLfloat position1 = 0.0f;  GLfloat position2 = 0.0f;  GLfloat position3 = 0.0f;  GLfloat position4 = 0.0f;  GLfloat speed = 0.05f;  void dis();  void display1();  void circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3f(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void update1(int value) {  if(position1 > 1.3)  position1 = 0.0f;  position1 += speed;  glutPostRedisplay();  glutTimerFunc(20,update1,0.0);  }  void update2(int value) {  if(position2 > 1.3)  position2 = 0.0f;  position2 += speed;  glutPostRedisplay();  glutTimerFunc(20,update2,0.0);  }  void update3(int value) {  if(position3 >1.3)  position3 =0.0f;  position3 += speed;  glutPostRedisplay();  glutTimerFunc(1500,update3,0);  }  void update4(int value) {  if(position4 < -1.3)  position4 = 0.0f;  position4 -= speed;  glutPostRedisplay();  glutTimerFunc(20,update4,0);  }  void init()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display1);  }  void display7()  {  //CIRCLE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  circle(0.2,0.0,0.0,1.0,0.0,0.0); // RED  glPopMatrix();  glutTimerFunc(1500,disback,0);  glFlush();  }  void display6(int val)  {  glutDisplayFunc(display7);  }  void display5()  {  //RECTANGLE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glBegin(GL\_POLYGON);  glColor3f(0.0f, 0.0f, 1.0f); //BLUE  glVertex2f(-0.2f, -0.1f);  glVertex2f( 0.2f, -0.1f);  glVertex2f( 0.2f, 0.1f);  glVertex2f(-0.2f, 0.1f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display6,0);  glFlush();  }  void display4(int val)  {  glutDisplayFunc(display5);  }  void display3()  {  // SQUARE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f); // GREEN  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 display1()  {  // HEXAGONE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f); //YELLOW  glVertex2f(0.2f,0.0f);  glVertex2f(0.1f, 0.2f);  glVertex2f(-0.1f, 0.2f);  glVertex2f(-0.2f, 0.0f);  glVertex2f(-0.1f, -0.2f);  glVertex2f(0.1f, -0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1500,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display1);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(850, 600);  glutInitWindowPosition(50, 50);  glutCreateWindow("4 Objects Calling");  glutDisplayFunc(dis);  init();  glutTimerFunc(20, update1, 0);  glutTimerFunc(20, update2, 0);  glutTimerFunc(20, update3, 0);  glutTimerFunc(20, update4, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-03**  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 <GL/glut.h>  #include<math.h>  using namespace std;  // MD. SHOHANUR RAHMAN SHOHAN  // 22-46013-1  GLfloat position1 = 0.0f;  GLfloat position2 = 0.0f;  GLfloat position3 = 0.0f;  GLfloat position4 = 0.0f;  GLfloat speed = 0.05f;  float \_move = 0.0f;  void dis();  void display1();  void circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3f(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void update1(int value)  {  if(position1 <-1.3)  position1 = 0.0f;  position1 -= speed;  glutPostRedisplay();  glutTimerFunc(100,update1,0.0);  }  void update2(int value)  {  if(position2 > 1.3)  position2 = 0.0f;  position2 += speed;  glutPostRedisplay();  glutTimerFunc(100,update2,0.0);  }  void update3(int value)  {  if(position3 >1.3)  position3 =0.0f;  position3 += speed;  glutPostRedisplay();  glutTimerFunc(100,update3,0);  }  void update4(int value)  {  if(position4 < -1.3)  position4 = 0.0f;  position4 -= speed;  glutPostRedisplay();  glutTimerFunc(100,update4,0);  }  void init()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  }  void disback(int val)  {  glutDisplayFunc(display1);  }  void display7()  {  // CIRCLE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f,position4, 0.0f);  circle(0.2,0.0,0.0,1.0,0.0,0.0); //RED  glPopMatrix();  glutTimerFunc(1000,disback,0);  glFlush();  }  void display6(int val)  {  glutDisplayFunc(display7);  }  void display5()  {  // RECTANGLE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(0.0f, position3, 0.5f);  glBegin(GL\_POLYGON);  glColor3f(0.0f, 0.0f, 1.0f); // BLUE  glVertex2f(-0.2f, -0.1f);  glVertex2f( 0.2f, -0.1f);  glVertex2f( 0.2f, 0.1f);  glVertex2f(-0.2f, 0.1f);  glEnd();  glPopMatrix();  glutTimerFunc(1000,display6,0);  glFlush();  }  void display4(int val)  {  glutDisplayFunc(display5);  }  void display3()  {  // SQUARE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position2, 0.0f, 0.0f);  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f); // GREEN  glVertex2f(-0.2f, -0.2f);  glVertex2f( 0.2f, -0.2f);  glVertex2f( 0.2f, 0.2f);  glVertex2f(-0.2f, 0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1000,display4,0);  glFlush();  }  void display2(int val)  {  glutDisplayFunc(display3);  }  void display1() {  // HEXAGONE  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glPushMatrix();  glTranslatef(position1, 0.0f, 0.0f);  glBegin(GL\_POLYGON);  glColor3f(1.0f, 1.0f, 0.0f); //YELLOW  glVertex2f(0.2f,0.0f);  glVertex2f(0.1f, 0.2f);  glVertex2f(-0.1f, 0.2f);  glVertex2f(-0.2f, 0.0f);  glVertex2f(-0.1f, -0.2f);  glVertex2f(0.1f, -0.2f);  glEnd();  glPopMatrix();  glutTimerFunc(1000,display2,0);  glFlush();  }  void dis()  {  glutDisplayFunc(display1);  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(850, 600);  glutInitWindowPosition(50, 50);  glutCreateWindow("4 Objects Moving");  glutDisplayFunc(dis);  init();  glutTimerFunc(200, update1, 0);  glutTimerFunc(200, update2, 0);  glutTimerFunc(200, update3, 0);  glutTimerFunc(200, update4, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-**    **Red Circle Moving Downwards**    **Blue Rectangle Moving Upward**    **Green Square Moving Right**    **Yellow Hexagon Moving Left** |

|  |
| --- |
| **Question-04**  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 <GL/glut.h>  #include<math.h>  using namespace std;  // MD. SHOHANUR RAHMAN SHOHAN  // 22-46013-1  float squareY = 0.0f;  float hexagonX = 0.0f;  float circleY = 0.0f;  float rectangleX = 0.0f;  void circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3f(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void display() {  // RECTANGLE  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3f(0.0f, 0.0f, 1.0f);  glBegin(GL\_POLYGON);  glVertex2f(rectangleX + 0.4f, -0.2f);  glVertex2f(rectangleX + 1.0f, -0.2f);  glVertex2f(rectangleX + 1.0f, 0.2f);  glVertex2f(rectangleX + 0.4f, 0.2f);  glEnd();  // CIRCLE  circle(0.2,0.0,circleY-0.6,1.0,0.0,0.0);  // HEXAGON  glColor3f(1.0f, 0.0f, 0.0f);  glBegin(GL\_POLYGON);  glVertex2f(hexagonX -0.8f, -0.2f);  glVertex2f(hexagonX -0.6, -0.2f);  glVertex2f(hexagonX -0.5f, 0.0f);  glVertex2f(hexagonX -0.6f, 0.2f);  glVertex2f(hexagonX -0.8f, 0.2f);  glVertex2f(hexagonX -0.9f, 0.0f);  glVertex2f(hexagonX -0.8f, -0.2f);  glEnd();  // SQUARE  glColor3f(0.0f, 1.0f, 0.0f);  glBegin(GL\_POLYGON);  glVertex2f(-0.2f, squareY + 0.4f);  glVertex2f(0.2f, squareY + 0.4f);  glVertex2f(0.2f, squareY + 0.8f);  glVertex2f(-0.2f, squareY + 0.8f);  glEnd();  glutSwapBuffers();  }  void handleKeyPress(unsigned char key, int x, int y) {  switch (key) {  case 'a': // Move to left  hexagonX -= 0.1f;  break;  case 'd': // Move to right  rectangleX += 0.1f;  break;  case 'w': // upward  squareY += 0.1f;  break;  case 's': //downward  circleY -= 0.1f;  break;  }  glutPostRedisplay();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(850, 600);  glutCreateWindow("4 Objects Moving by Keyboard");  glutDisplayFunc(display);  glutKeyboardFunc(handleKeyPress);  gluOrtho2D(-2,2,-2,2);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-05**  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 <GL/glut.h>  #include<math.h>  using namespace std;  // MD. SHOHANUR RAHMAN SHOHAN  // 22-46013-1  float squareX = 0.0f;  float hexagonX = 0.0f;  float circleX = 0.0f;  float rectangleX = 0.0f;  void circle(float radius, float xc, float yc, float r, float g, float b)  {  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3f(r,g,b);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=radius;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+xc,y+yc);  }  glEnd();  }  void display() {  // RECTANGLE  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3f(0.0f, 0.0f, 1.0f);  glBegin(GL\_POLYGON);  glVertex2f(rectangleX + 0.4f, -0.2f);  glVertex2f(rectangleX + 1.0f, -0.2f);  glVertex2f(rectangleX + 1.0f, 0.2f);  glVertex2f(rectangleX + 0.4f, 0.2f);  glEnd();  // CIRCLE  circle(0.2,circleX +0.0,-0.6,1.0,0.0,0.0);  // HEXAGON  glColor3f(1.0f, 0.0f, 0.0f);  glBegin(GL\_POLYGON);  glVertex2f(hexagonX -0.8f, -0.2f);  glVertex2f(hexagonX -0.6, -0.2f);  glVertex2f(hexagonX -0.5f, 0.0f);  glVertex2f(hexagonX -0.6f, 0.2f);  glVertex2f(hexagonX -0.8f, 0.2f);  glVertex2f(hexagonX -0.9f, 0.0f);  glVertex2f(hexagonX -0.8f, -0.2f);  glEnd();  // SQUARE  glColor3f(0.0f, 1.0f, 0.0f);  glBegin(GL\_POLYGON);  glVertex2f(squareX -0.2f,0.4f);  glVertex2f(squareX +0.2f,0.4f);  glVertex2f(squareX +0.2f,0.8f);  glVertex2f(squareX -0.2f,0.8f);  glEnd();  glutSwapBuffers();  }  void handleMouse(int button, int state, int x, int y) {  if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN) {  squareX -= 0.1f;  hexagonX -= 0.1f;  } else if (button == GLUT\_RIGHT\_BUTTON && state == GLUT\_DOWN) {  rectangleX += 0.1f;  circleX += 0.1f;  }  glutPostRedisplay();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitWindowSize(850, 600);  glutCreateWindow("4 Objects Moving by Keyboard");  glutDisplayFunc(display);  glutMouseFunc(handleMouse);  gluOrtho2D(-2,2,-2,2);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |