Assignment -3

/*Problem Statement: Implement Bresenham circle drawing algorithm to draw any object. The object should be displayed in all the quadrants with respect to centre and radius */

```
 #include<GL/glut.h>
#include<GL/glu.h>
#include<iostream>
#include<math.h>
using namespace std;
GLint xi,yi,r,choice=1;
void setpixel(GLint x, GLint y)
{
   if(choice==1)
   {
      glColor3f(1.0,0.0,0.0);
   }
   else if(choice==2)
   {
      glColor3f(0.0,0.0,0.0);
   }
   else if(choice==3)
   {
      glColor3f(1.0,1.0,0.0);
   }
   else if(choice==4)
   {
      glColor3f(0.4,0.9,0.2);
   }
   else if(choice==5)
   {
      glColor3f(0.0,0.0,1.0);
   }
   glBegin(GL_POINTS);
   glVertex2f(x,y);
   glEnd();
   glFlush();
}
void myDisplay(void)
{
   /*glClear(GL_COLOR_BUFFER_BIT);
   glFlush();*/
}
void drawCircle(int xc, int yc, int x, int y)
{
   setpixel(xc+x, yc+y);
   setpixel(xc-x, yc+y);
   setpixel(xc+x, yc-y);
   setpixel(xc-x, yc-y);
   setpixel(xc-y, yc+x);
   setpixel(xc-y, yc-x);
```

```
   setpixel(xc+y, yc+x);
   setpixel(xc+y, yc-x);
}
void initialize()
{
   glClearColor(0.6,0.6,0.6,0.0);
   glClear(GL_COLOR_BUFFER_BIT);
   //glColor3f(1.0f,0.0f,0.0f);
   glPointSize(5.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0.0,640.0,0.0,480.0);
}

class Circle
{
public:
   void BresCircle()
   {
      cout<<"Enter the value of radius  :";
      cin>>r;
      GLint x = 0, y = r;
      GLint s = 3 - 2 * r;
      drawCircle(xi, yi, x, y);
      while (y >= x)
      {
         x++;
         if (s > 0)
         {
            y--;
            s = s + 4 * (x - y) + 10;
         }
         else
            s = s + 4 * x + 6;
         drawCircle(xi, yi, x, y);
      }
   }
};
Circle l;
void keyboard(unsigned char key, int x, int y)
{
   if(key==27)
      exit(0); }
void mouse( int button, int state, int x, int y )
{
   if( state == GLUT_DOWN )
   {
      if( button == GLUT_LEFT_BUTTON)
      {
         xi=x;
         yi=480-y;
```

```
        cout<<xi<<"\t";
        cout<<yi<<"\n";
        glPointSize(2.0);
        glBegin(GL_POINTS);
        glVertex2i(xi,yi);
        glEnd();
        glFlush();
      }   }  }
void menu(int item)
{
   choice = item;
   if(item==1 || item==2 || item==3 || item==4 || item==5)
   {
      l.BresCircle();
   }
   if(item==6)
   {
      exit(0);
   }
}

int main(int argc, char **argv)
{
   glutInit(&argc,argv);
   glutInitWindowSize(640,480);
   glutInitWindowPosition(100,0);
   glutCreateWindow("Bresenham circle Drawing!!");
   initialize();
   glutDisplayFunc(myDisplay);
   glutMouseFunc(mouse);
   glutKeyboardFunc(keyboard);
   glutCreateMenu(menu);
   glutAddMenuEntry("Draw Red CIRCLE",1);
   glutAddMenuEntry("Draw Black CIRCLE",2);
   glutAddMenuEntry("Draw Yellow CIRCLE",3);
   glutAddMenuEntry("Draw Green CIRCLE",4);
   glutAddMenuEntry("Draw Blue CIRCLE",5);
   glutAddMenuEntry("EXIT",6);
   glutAttachMenu(GLUT_MIDDLE_BUTTON);
   glutMainLoop();
   return 0;
}
```