Assignment No. 3

Title	Pattern drawing using line and circle.
Aim/Problem Statement	Write C++ program to draw a given pattern. Use DDA line and Bresenham's circle drawing algorithm. Apply the concept of encapsulation.
CO Mapped	
Pre-requisite	1. Basic programming skills of C++ 2. 64-bit Open source Linux 3. Open Source C++ Programming tool like G++/GCC
Learning Objective	To learn and apply DDA line and Bresenham's circle drawing algorithm.

Theory:

DDA Line Drawing Algorithm:

Line is a basic element in graphics. To draw a line, you need two end points between which you can draw a line. Digital Differential Analyzer (DDA) line drawing algorithm is the simplest line drawing algorithm in computer graphics. It works on incremental method. It plots the points from starting point of line to end point of line by incrementing in X and Y direction in each iteration.

DDA line drawing algorithm works as follows:

Step 1: Get coordinates of both the end points (X_1, Y_1) and (X_2, Y_2) from user.

Step 2: Calculate the difference between two end points in X and Y direction.

```
dx = X2 – X1;
dy = Y2 – Y1;
```

Step 3: Based on the calculated difference in step-2, you need to identify the number of steps to put pixel. If dx > dy, then you need more steps in x coordinate; otherwise in y coordinate.

```
if (absolute(dx) > absolute(dy))
        Steps = absolute(dx);
else
        Steps = absolute(dy);
```

Step 4: Calculate the increment in x coordinate and y coordinate.

```
Xincrement = dx / (float) steps;
Yincrement = dy / (float) steps;
```

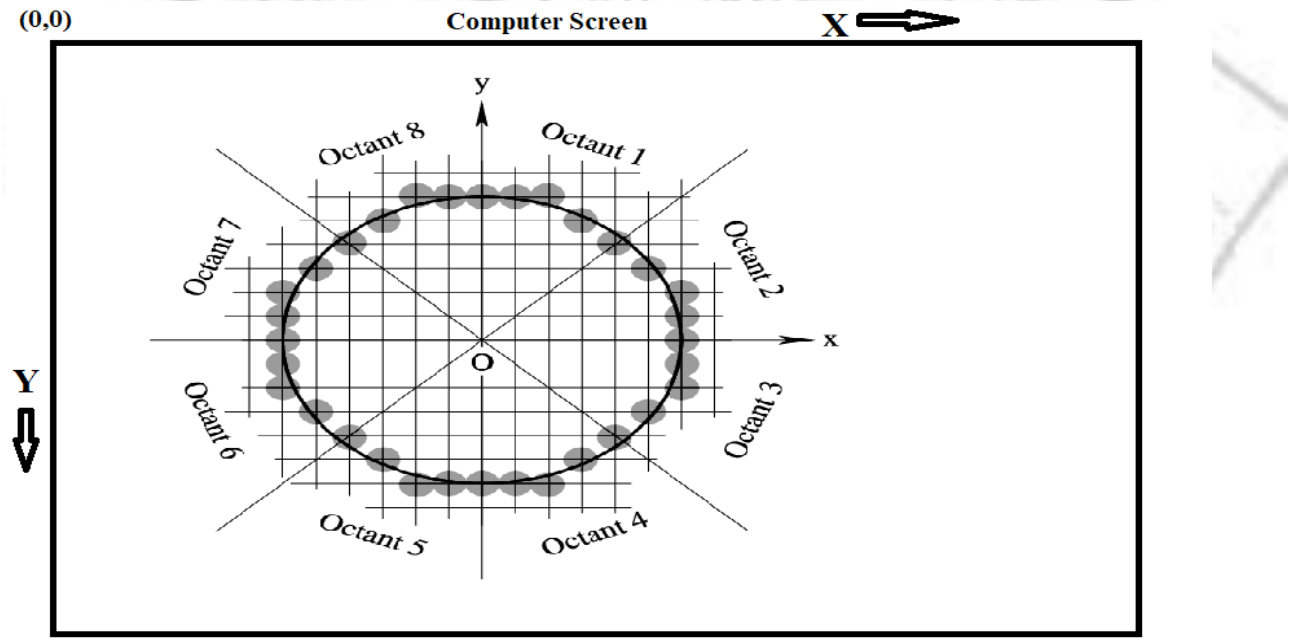
Step 5: Plot the pixels by successfully incrementing x and y coordinates accordingly and complete the drawing of the line.

```
for(int i=0; i < Steps; i++)
{
  X1 = X1 + Xincrement;
  Y1 = Y1 + Yincrement;
  putpixel(Round(X1), Round(Y1), ColorName);
}
```

Bresenham's Circle Drawing Algorithm:

Circle is an eight-way symmetric figure. The shape of circle is the same in all quadrants. In each quadrant, there are two octants. If the calculation of the point of one octant is done, then the other seven points can be calculated easily by using the concept of eight-way symmetry.

Bresenham's Circle Drawing Algorithm is a circle drawing algorithm that selects the nearest pixel position to complete the arc. The unique part of this algorithm is that is uses only integer arithmetic which makes it significantly faster than other algorithms using floating point arithmetic.



Step 1: Read the x and y coordinates of center: (centx, centy)

Step 2: Read the radius of circle: (r)

Step 3: Initialize, x = 0; y = r;

Step 4: Initialize decision parameter: p = 3 – (2 * r)

Step 5:

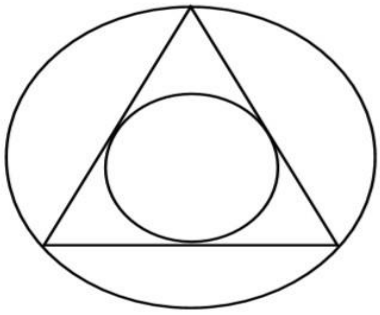
```
do {
        putpixel(centx+x, centy-y, ColorName);
        if (p<0)
                p = p+(4*x)+6;
        else
                {
                p=p+[4*(x-y)]+10;
                y=y-1;
                }
        x=x+1;
        } while(x<y)
```

Here in step 5, putpixel() function is used which will print Octant-1 of the circle with radius r after the completion of all the iterations of do-while loop. Because of the eight-way symmetry property of circle, we can draw other octants of the circle using following putpixel() functions:

Octant 1: putpixel(centx+x, centy-y, ColorName);
Octant 2: putpixel(centx+y, centy-x, ColorName);

Octant 3: putpixel(centx+y, centy+x, ColorName);
Octant 4: putpixel(centx+x, centy+y, ColorName);
Octant 5: putpixel(centx-x, centy+y, ColorName);
Octant 6: putpixel(centx-y, centy+x, ColorName);
Octant 7: putpixel(centx-y, centy-x, ColorName);
Octant 8: putpixel(centx-x, centy-y, ColorName);

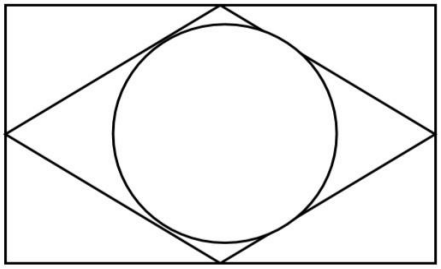
Drawing Pattern using lines and circles:



This pattern is made up of one equilateral triangle and two concentric circles. To draw the triangle, we require coordinates of 3 vertices forming an equilateral triangle. To draw two concentric circles, we require coordinates of common center and radius of both the circles.

We will take coordinates of circle and radius of one of the circle from user. Then using the properties of an equilateral triangle, we can find all 3 vertices of equilateral triangle and radius of other circle. Once we get all these parameters, we can call DDA line drawing and Bresenham's circle drawing algorithms by passing appropriate parameters to get required pattern.

OR



To draw this pattern, we require two rectangles and one circle. We can use suitable geometry to get coordinates to draw rectangles and circle. Then we can call DDA line drawing and Bresenham's circle drawing algorithms by passing appropriate parameters to get required pattern.

Conclusion:

Questions:

1. Explain the derivation of decision parameters in Bresenham's circle drawing algorithm.
2. Explain the concept of encapsulation with example.