**Assignment-01**

# Pattern drawing using DDA line and Bresenham circle drawing algorithm (concept of encapsulation)

## Pattern 1

```
#include<iostream>
#include<graphics.h>
#include<math.h>
using namespace std;
class line
{

 int x,y,dx,dy,xi,yi,steps;
 public:

void put()
{
putpixel(x,y,9);
}
```

```
void lin(int x1,int y1,int x2, int y2)
{

dx=x2-x1;
dy=y2-y1;
if(abs(dx)>abs(dy))
 steps=abs(dx);
else
steps=abs(dy);

xi=dx/steps;
yi=dy/steps;

x=x1+0.5;
y=y1+0.5;
put();

for (int  i=1;i<=steps;i++)
{
x=x+xi;
y=y+yi;
put();
```

```
delay(5);
}
}
}l;

class circle1
{
public :

void circle(int x,int y,intrad)
{
int x1=0;
int y1=rad;
putpixel(x1,y1,6);
int pk=3-2*rad;
while(x1<y1)
{
if(pk<=0)
{
pk=pk+(4*x1)+6;
}
else
```

```
{
pk=pk+4*(x1-y1)+10;
y1--;
}
x1++;
putpixel(x+x1,y+y1,3);
putpixel(x-x1,y+y1,3);
putpixel(x+x1,y-y1,3);
putpixel(x-x1,y-y1,3);
putpixel(x+y1,y+x1,3);
putpixel(x-y1,y+x1,3);
putpixel(x+y1,y-x1,3);
putpixel(x-y1,y-x1,3);
delay(10);
}
}
}c;
int main()
{
int gd=DETECT,gm=DETECT;
initgraph(&gd,&gm,NULL);
```

```
l.lin(100,100,400,100);
l.lin(100,400,400,400);
l.lin(100,100,100,400);
l.lin(400,100,400,400);
l.lin(250,100,100,250);
l.lin(100,250,250,400);
l.lin(250,100,400,250);
l.lin(400,250,250,400);
/*l.lin(175,175,325,175);
l.lin(175,325,325,325);
l.lin(175,175,175,325);
l.lin(325,175,325,325);*/
c.circle(250,250,100);
delay(500);
closegraph();
return 0;
}
```

The provided code is a C++ program that uses the graphics.h library to draw lines and a circle on the screen. Here’s a

breakdown of its key components and functionality:

Explanation

Classes:

line: This class handles line drawing using the Bresenham'sline algorithm.

put(): Places a pixel on the screen at the current coordinates.

lin(): Calculates the increments for x and y coordinates and iteratively draws a line from (x1, y1) to (x2, y2).

circle1: This class handles drawing circles using Bresenham's circle algorithm.

circle(): Draws a circle centered at (x, y) with a specified radius.

Main Function:

Initializes the graphics mode.

Creates various lines to form a shape.

Draws a circle at the center of the shape.

Waits for a brief moment and then closes the graphics window.

Summary

This program is a simple graphics application that demonstrates line and circle drawing using pixel manipulation. The use of putpixel() allows it to visualize shapes, and the delay provides a slight pause to make the drawing visible. Ensure that your development environment supports the graphics.h library to run this code.