12/25/22, 2:56 PM A1.cpp

```
1 /*Write C++ program to draw the following pattern.Use DDA line and
 2 Bresenham drawing algorithm. Apply the concept of encapsulation.*/
 3 #include <iostream>
 4 #include <graphics.h>
 5 #include <bits/stdc++.h>
       using namespace std;
 7 class algo
 8 {
 9 public:
10
       void dda_line(float x1, float y1, float x2, float y2);
       void bresneham_cir(int r);
11
12 };
13 void algo::dda_line(float x1, float y1, float x2, float y2)
14 {
15
       float x, y, dx, dy, step;
16
       int i;
17
       // step 2
       dx = abs(x2 - x1);
18
19
       dy = abs(y2 - y1);
       cout << "dy=" << dy << "\tdx=" << dx;
20
21
       // step 3
22
       if (dx >= dy)
23
           step = dx;
24
       else
25
           step = dy;
       cout << "\n"
26
27
            << step << endl;
28
       // step 4
29
       float xinc = float((x2 - x1) / step);
30
       float yinc = float((y2 - y1) / step);
31
       // step 5
32
       x = x1;
33
       y = y1;
       // outtextxy(0,0,"(0,0)");
34
35
       // step 6
       i = 1;
36
37
       while (i <= step)
38
           // cout<<endl<<"\t"<<i<<"\t(x,y)=("<<x<<","<<y<<")";
39
40
           putpixel(320 + x, 240 - y, 4);
41
           x = x + xinc;
42
           y = y + yinc;
43
           i = i + 1;
44
           // delay(10);
45
       }
46 }
47 void algo::bresneham cir(int r)
48 {
       float x, y, p;
49
50
       x = 0;
51
       y = r;
52
       p = 3 - (2 * r);
53
       while (x <= y)
54
55
           putpixel(320 + x, 240 + y, 1);
56
           putpixel(320 - x, 240 + y, 2);
           putpixel(320 + x, 240 - y, 3);
57
           putpixel(320 - x, 240 - y, 5);
58
59
           putpixel(320 + y, 240 + x, 6);
```

localhost:4649/?mode=clike

12/25/22, 2:56 PM A1.cpp

```
60
           putpixel(320 + y, 240 - x, 7);
           putpixel(320 - y, 240 + x, 8);
61
62
           putpixel(320 - y, 240 - x, 9);
63
           x = x + 1;
           if (p < 0)
64
65
                p = p + 4 * (x) + 6;
66
67
           }
68
           else
69
           {
70
                p = p + 4 * (x - y) + 10;
71
               y = y - 1;
72
73
           // delay(20);
74
       }
75 }
76 int main()
77 {
78
       algo a1;
79
       int i;
80
       float r, ang, r1;
81
       initwindow(630, 480);
82
       cout << "Enter radius of circle";</pre>
83
       cin >> r;
       a1.bresneham_cir((int)r);
84
85
       ang = 3.24 / 180;
       float c = r * cos(30 * ang);
86
87
       float s = r * sin(30 * ang);
88
       a1.dda_line(0, r, 0 - c, 0 - s);
       a1.dda_line(0 - c, 0 - s, 0 + c, 0 - s);
89
90
       a1.dda_line(0 + c, 0 - s, 0, r);
91
       r1 = s;
92
       a1.bresneham_cir((int)r1);
93
       getch();
94
       closegraph();
95
       return 0;
96 }
```

localhost:4649/?mode=clike 2/2

