```
//PROGRAM: To implement Bresenham's Circle Generation Algorithm
#include<iostream>
#include<graphics.h>
using namespace std;

class base //BASE CLASS START
{
 public:
  void display(int x, int y, int xc, int yc)
  {
   putpixel(xc+x,yc+y,WHITE);delay(10);
   putpixel(xc+x,yc-y,WHITE);delay(10);
   putpixel(xc-x,yc+y,WHITE);delay(10);
   putpixel(xc-x,yc-y,WHITE);delay(10);
   putpixel(xc+y,yc+x,WHITE);delay(10);
   putpixel(xc+y,yc-x,WHITE);delay(10);
   putpixel(xc-y,yc+x,WHITE);delay(10);
   putpixel(xc-y,yc-x,WHITE);
  }
};//BASE CLASS END

class pixel : public base
{
 public:

  void bresencircle(int xc, int yc, int r)
  {
   int x=0,y=r;
   float dp=3-(2*r);

   putpixel(x,y,WHITE);

   while(x<=y)
   {
    if(dp<=0)
     {
      dp=dp+(4*x)+6;
      x++;
     }
    else
     {
      dp=dp+(4*(x-y))+10;
      x++;
      y--;
     }

    display(x,y, xc, yc);
   };
  }
}obj;


int main()
{
 int xc,yc,r;

 cout<<"\n Enter the center coordinates and radius (xc,yc,r): ";
 cin>>xc>>yc>>r;

 int gd=DETECT,gm;
 initgraph(&gd,&gm, NULL);

 obj.bresencircle(xc,yc,r);

 getch();
 closegraph();
```

```
 return(0);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
                                                OUTPUT
gauravgarje@gaurav-Inspiron-3542:~$ g++ cgprac5.cpp -lgraph
gauravgarje@gaurav-Inspiron-3542:~$ ./a.out

 Enter the center coordinates and radius (xc,yc,r): 300
250
80
gauravgarje@gaurav-Inspiron-3542:~$
```