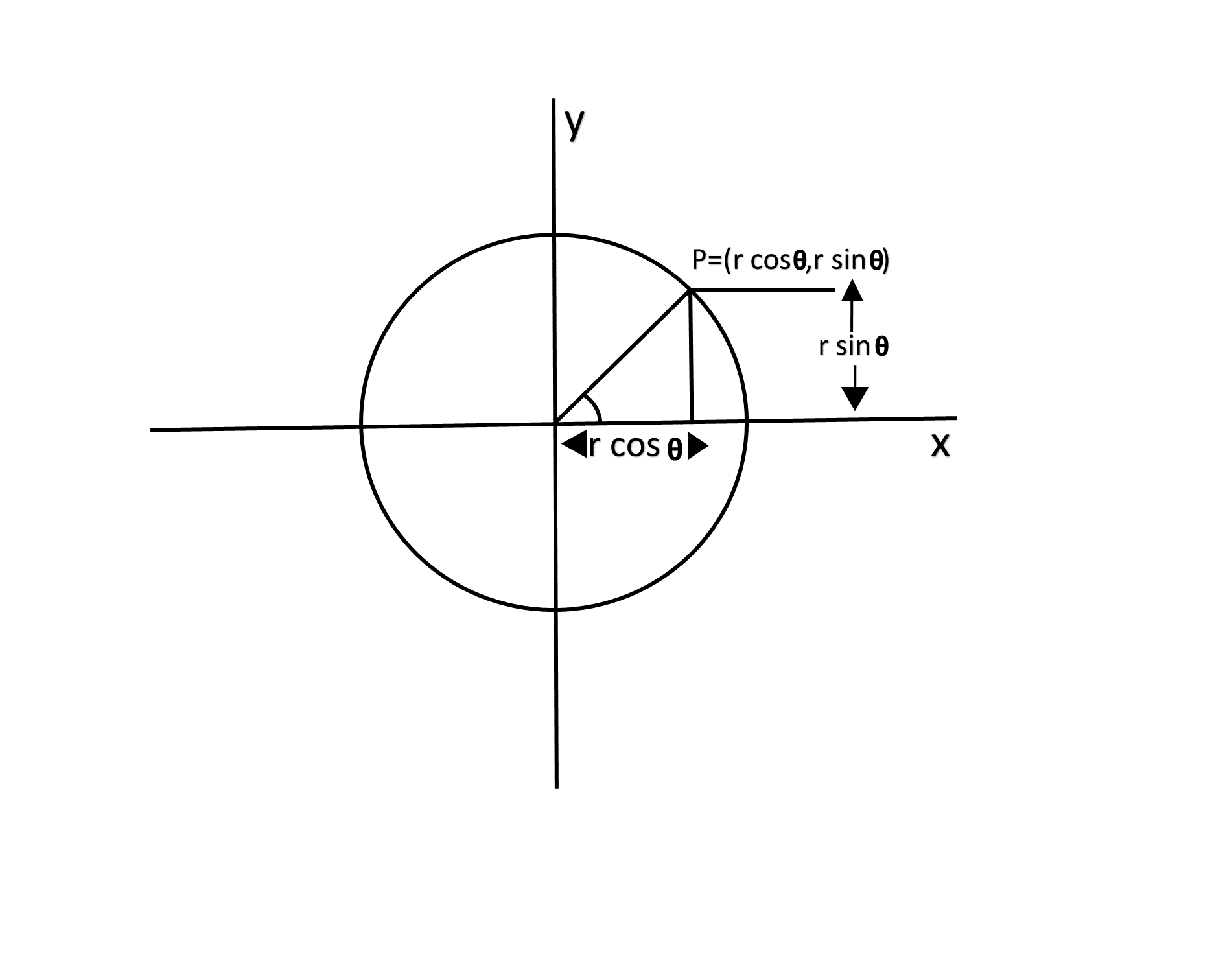
Defining a circle using Polar Co-ordinates :

The second method of defining a circle makes use of polar coordinates as shown in fig:

            x=r cos θ             y = r sin θ  
Where θ=current angle  
r = circle radius  
x = x coordinate  
y = y coordinate

By this method, θ is stepped from 0 to Defining a circle using Polar Co-ordinates & each value of x & y is calculated.



Algorithm:

**Step1:** Set the initial variables:

            r = circle radius  
            (h, k) = coordinates of the circle center  
                i = step size  
            θ\_end=Defining a circle using Polar Co-ordinates  
            θ=0

**Step2:** If θ>θendthen stop.

**Step3:** Compute

            x = r \* cos θ            y=r\*sin?θ

**Step4:** Plot the eight points, found by symmetry i.e., the center (h, k), at the current (x, y) coordinates.

Plot (x + h, y +k)             Plot (-x + h, -y + k)  
Plot (y + h, x + k)             Plot (-y + h, -x + k)  
Plot (-y + h, x + k)             Plot (y + h, -x + k)  
Plot (-x + h, y + k)             Plot (x + h, -y + k)

**Step5:** Increment θ=θ+i

**Step6:** Go to step (ii).

Program to draw a circle using Polar Coordinates:

1. #include <graphics.h>
2. #include <stdlib.h>
3. #define color 10
4. **void** eightWaySymmetricPlot(**int** xc,**int** yc,**int** x,**int** y)
5. {
6. putpixel(x+xc,y+yc,color);
7. putpixel(x+xc,-y+yc,color);
8. putpixel(-x+xc,-y+yc,color);
9. putpixel(-x+xc,y+yc,color);
10. putpixel(y+xc,x+yc,color);
11. putpixel(y+xc,-x+yc,color);
12. putpixel(-y+xc,-x+yc,color);
13. putpixel(-y+xc,x+yc,color);
14. }
15. **void** PolarCircle(**int** xc,**int** yc,**int** r)
16. {
17. **int** x,y,d;
18. x=0;
19. y=r;
20. d=3-2\*r;
21. eightWaySymmetricPlot(xc,yc,x,y);
22. **while**(x<=y)
23. {
24. **if**(d<=0)
25. {
26. d=d+4\*x+6;
27. }
28. **else**
29. {
30. d=d+4\*x-4\*y+10;
31. y=y-1;
32. }
33. x=x+1;
34. eightWaySymmetricPlot(xc,yc,x,y);
35. }
36. }
37. **int** main(**void**)
38. {
39. **int** gdriver = DETECT, gmode, errorcode;
40. **int** xc,yc,r;
41. initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
42. errorcode = graphresult();
43. **if** (errorcode != grOk)
44. {
45. printf("Graphics error: %s\n", grapherrormsg(errorcode));
46. printf("Press any key to halt:");
47. getch();
48. exit(1);
49. }
50. printf("Enter the values of xc and yc ,that is center points of circle : ");
51. scanf("%d%d",&xc,&yc);
52. printf("Enter the radius of circle : ");
53. scanf("%d",&r);
54. PolarCircle(xc,yc,r);
55. getch();
56. closegraph();
57. **return** 0;
58. }

**Output:**

