- Find out the pixels required to draw a circle with radius of 15 and centered at origin

radius = 15

x	y	d	E/SE	d update

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14		

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	**E**	

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	**=-14+2x0+3 = -11**

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	**15**			

```
func MidpointCircle(int radius, int value){
    int x, y, d;
    d = 1 - radius;
    x = 0;
    y = radius;
    Circlepoints(x, y, value);
    while (x < y) {
        if (d < 0) {
            //choose E
            d = d + 2*x + 3;
            x = x + 1;
        }
        else {
            //choose SE
            d = d + 2*x – 2*y + 5;
            x = x + 1;
            y = y - 1;
        }
        Circlepoints(x,y, value)
    }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	15	**-11**	**E**	

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	**=-11+2x1+3=-6**

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3=-6
2	**15**			

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3=-6
2	15	**-6**	**E**	

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	**E**	**=-6+2x2+3= 1**

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	**15**			

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	**1**	**SE**	

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	**=1+2x3-2x15+5= -18**

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	**14**			

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	14	**-18**	**E**	

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	14	-18	E	**=-18+2x4+3 = -7**

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	14	-18	E	=-18+2x4+3 = -7
5	**14**			

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	**y**	**d**	**E/SE**	**d update**
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	14	-18	E	=-18+2x4+3 = -7
5	14	**-7**	**E**	**=-7+2x5+3= 6**
6	**14**			

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	14	-18	E	=-18+2x4+3 = -7
5	14	-7	E	=-7+2x5+3= 6
6	14	**6**	**SE**	**=6+2x6-2x14+5= -5**
7	**13**			

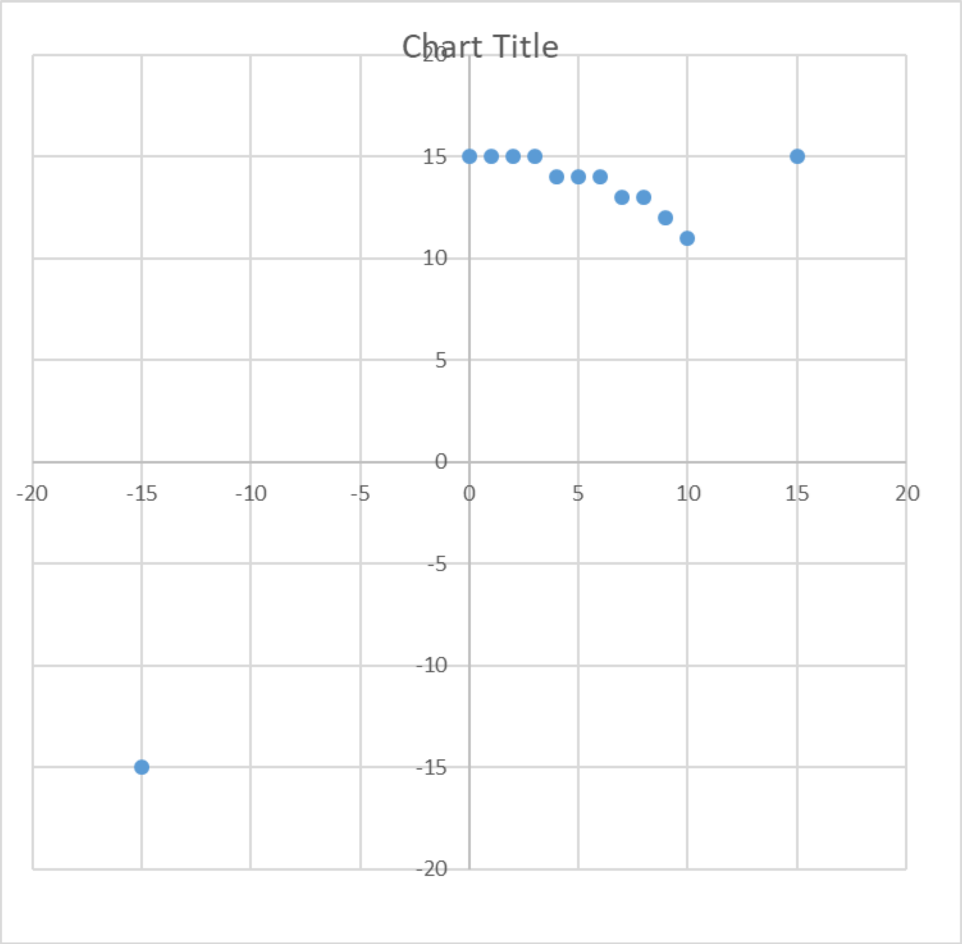
```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

radius = 15

x	y	d	E/SE	d update
0	15	-14	E	=-14+2x0+3 = -11
1	15	-11	E	=-11+2x1+3= -6
2	15	-6	E	=-6+2x2+3= 1
3	15	1	SE	=1+2x3-2x15+5= -18
4	14	-18	E	=-18+2x4+3 = -7
5	14	-7	E	=-7+2x5+3= 6
6	14	6	SE	=6+2x6-2x14+5= -5
7	13	-5	E	=-5+2x7+3= 12
8	13	12	SE	=12+2x8-2x13+5= 7
9	12	7	SE	=7+2x9-2x12+5= 6
10	11	6	SE	=6+2x10-2x11+5= 8
11	10			

```
func MidpointCircle(int radius, int value){
        int x, y, d;
        d = 1 - radius;
        x = 0;
        y = radius;
        Circlepoints(x, y, value);
        while (x < y) {
                if (d < 0) {
                        //choose E
                        d = d + 2*x + 3;
                        x = x + 1;
                }
                else {
                        //choose SE
                        d = d + 2*x – 2*y + 5;
                        x = x + 1;
                        y = y - 1;
                }
                Circlepoints(x,y, value)
        }
}
```

x	y
0	15
1	15
2	15
3	15
4	14
5	14
6	14
7	13
8	13
9	12
10	11



Zone 1			Zone 6			Zone 5			Zone 2			Zone 0			Zone 3			Zone 4			Zone 7	
(X, Y)			(X, -Y)			(-X,- Y)			(-X, Y)			(Y, X)			(-Y, X)			(-Y, -X)			(Y, -X)	
0	15		0	-15		0	-15		0	15		15	0		-15	0		-15	0		15	0
1	15		1	-15		-1	-15		-1	15		15	1		-15	1		-15	-1		15	-1
2	15		2	-15		-2	-15		-2	15		15	2		-15	2		-15	-2		15	-2
3	15		3	-15		-3	-15		-3	15		15	3		-15	3		-15	-3		15	-3
4	14		4	-14		-4	-14		-4	14		14	4		-14	4		-14	-4		14	-4
5	14		5	-14		-5	-14		-5	14		14	5		-14	5		-14	-5		14	-5
6	14		6	-14		-6	-14		-6	14		14	6		-14	6		-14	-6		14	-6
7	13		7	-13		-7	-13		-7	13		13	7		-13	7		-13	-7		13	-7
8	13		8	-13		-8	-13		-8	13		13	8		-13	8		-13	-8		13	-8
9	12		9	-12		-9	-12		-9	12		12	9		-12	9		-12	-9		12	-9
10	11		10	-11		-10	-11		-10	11		11	10		-11	10		-11	-10		11	-10

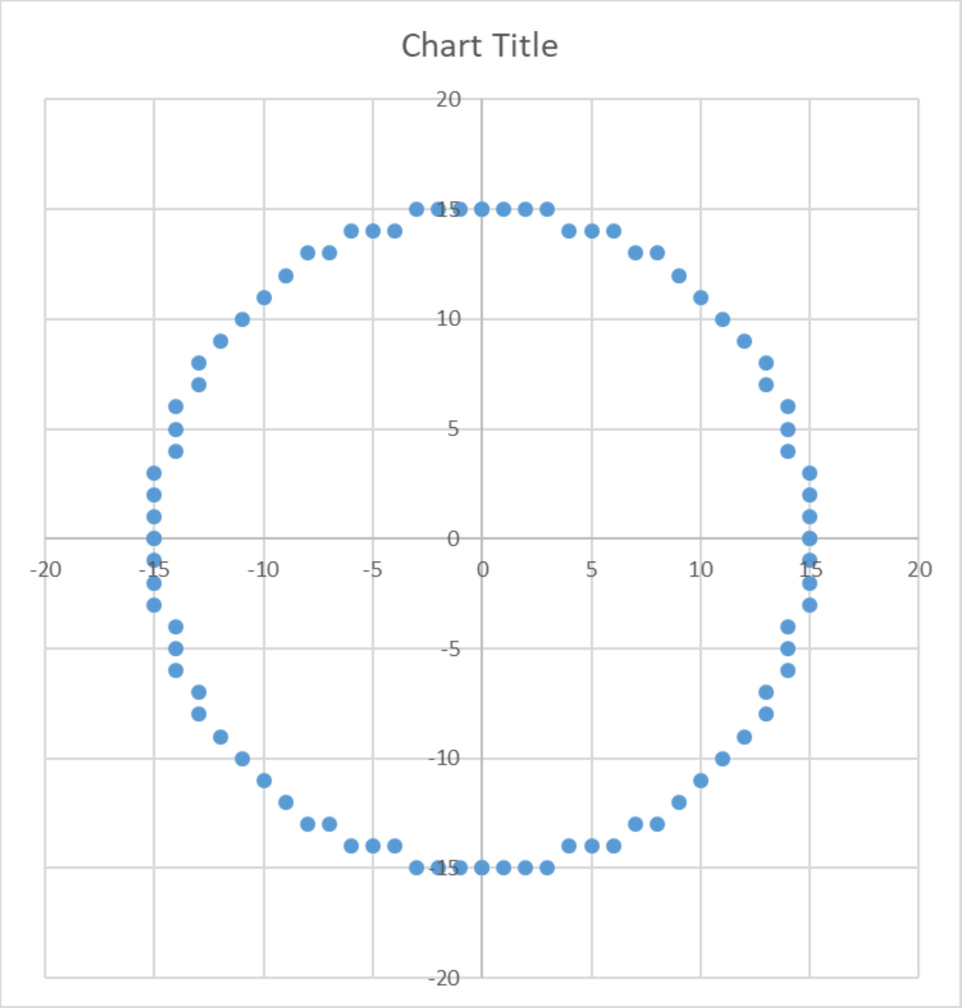


Chart Title
20
15
10
5
0
-5
-10
-15
-20
-20
-15
-10
-5
0
5
10
15
20

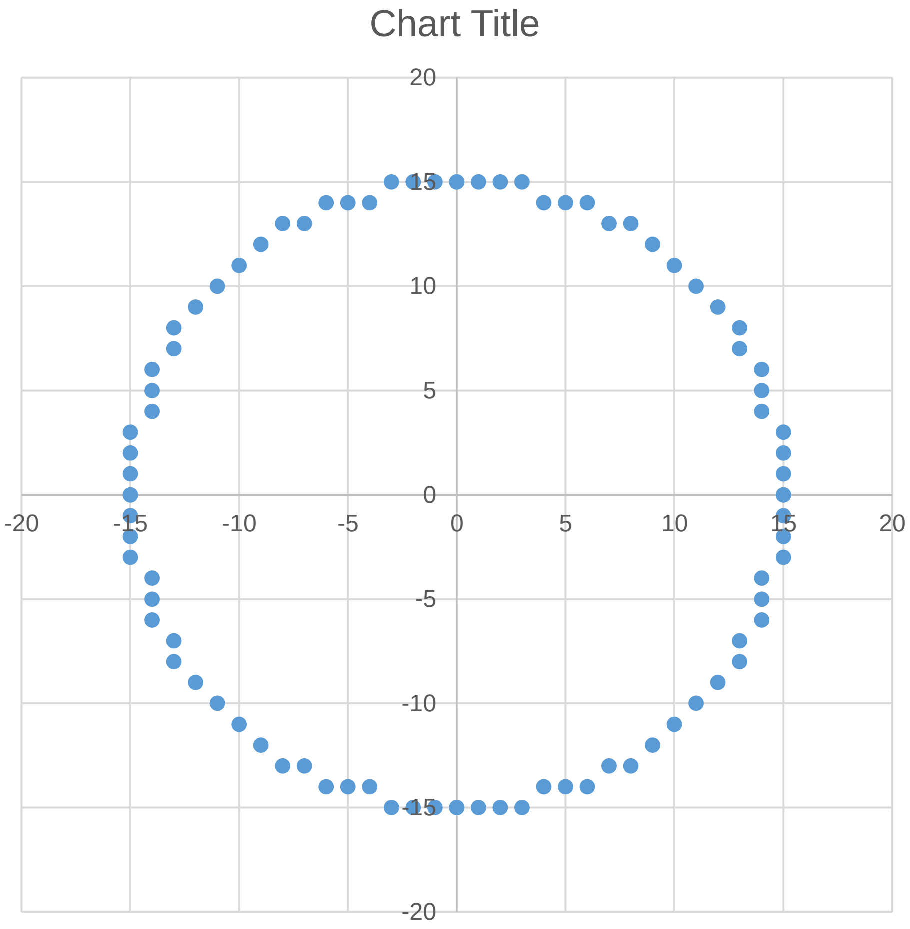


Chart Title
20
15
10
5
0
-5
-10
-15
-20
-20
-15
-10
-5
0
5
10
15
20