Graphics in C language

Lecture by Waseemullah

Lecturer

Department of CS&IT

NED University of Engineering and Technology, Karachi

Text mode vs Graphics mode of C

Text Mode	Graphics Mode
Basic Unit is character	Basic Unit is pixel
80 columns, 50 rows	640 columns, 480 rows

My First program/drawing a line

```
#include<conio.h>
#include<graphics.h>

main()
    {
    int gd=DETECT, gm;

    initgraph(&gd, &gm, “C:/TC/bgi”);

    line(0,0,200,200);

    getch();
    closegraph();
    }
```

My First program/drawing a line

Add Graphics header file at the top

```
#include<conio.h>
#include<graphics.h>
```

Define two variables for Graphics Driver and Graphics mode.

```
main()
    {
    int gd=DETECT, gm;
```

Initialize the Graphics mode

Path of the bgi folder.

```
    initgraph(&gd, &gm, “C:/TC/bgi”);
```

Starting x and y co-ordinates of the line.

```
    line(0,0,200,200);
```

Ending x and y co-ordinates of the line.

```
    getch();
    closegraph();
    }
```

Close Graphics mode.

Drawing a Circle

```
#include<conio.h>
#include<graphics.h>

main()
	{
	int gd=DETECT, gm;

	initgraph(&gd, &gm, “C:/TC/bgi”);

	circle(320,240,100);

	getch();
	closegraph();
	}
```

circle(320,240,100);

Explanation

- A Circle takes a total of 3 arguments.
- The first two arguments are used to define center of the circle in x and y co-ordinates.
- Since screen has a size of 640 pixels in x-axis, so 320 is the center of x-axis.
- And screen has the size of 480 pixels in y-axis, so 240 is the center of y-axis.
- Third argument of the circle is its radius in pixels. In our example the radius of the circle is 100 pixels.

Some other shapes in graphics.h library.

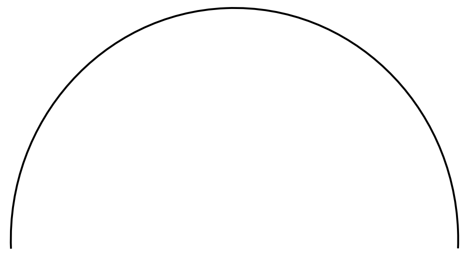
arc(midx, midy,starting-angle,ending-angle,radius);

arc(midx, midy,starting-angle,ending-angle,radius);

- Explanation:
  - Arc is used to draw circular arc
  - Arc takes 5 arguments, all of the int type.
  - First two arguments define the center of the arc to place on the screen.
  - Third and Fourth arguments are starting and ending angles of the arc.
  - Fifth argument is the radius of the arc in pixels.

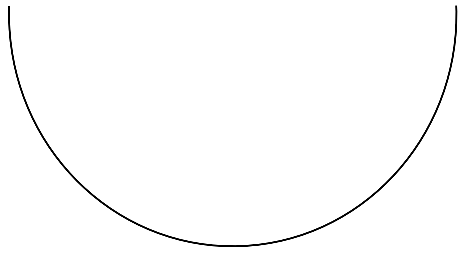
arc(midx, midy,starting-angle,ending-angle,radius);

- arc(320,240,0,180,100);
  - The above code generates an arc in the mid of the screen, angled from 0 to 180(making a half circle facing downwards), having radius of the 100 pixels.



arc(midx, midy,starting-angle,ending-angle,radius);

- arc(320,240,180,0,100);
  - The above code generates an arc in the mid of the screen, angled from 0 to 180(making a half circle facing upwards), having radius of the 100 pixels.



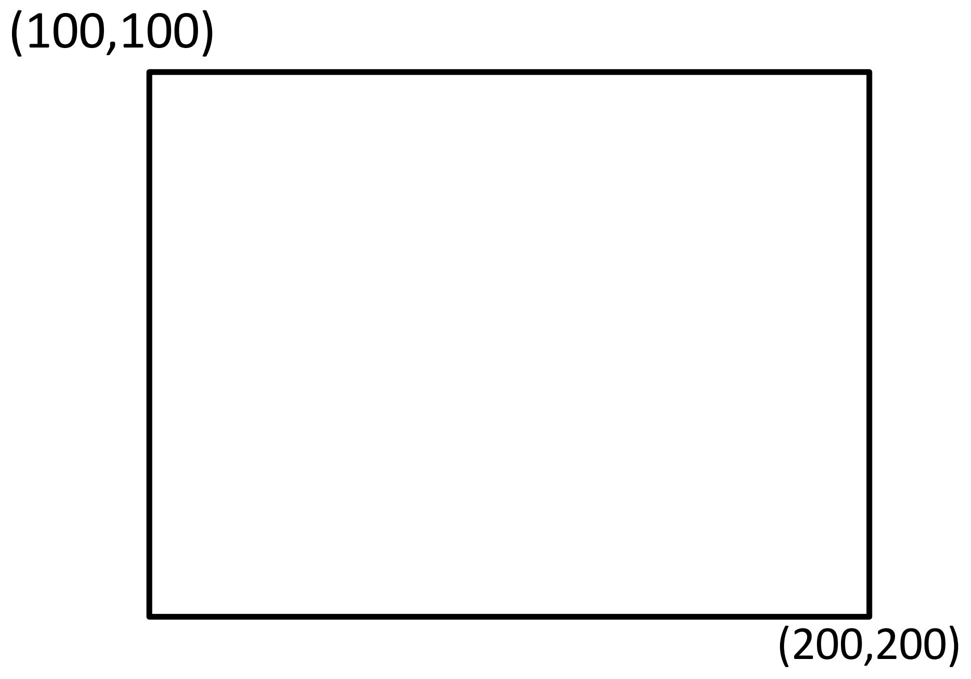
ractangle(left, top,right, bottom);

ractangle(left, top,right, bottom);

- Explanation:
  - Rectangle is used to draw an empty rectangle.
  - It takes 4 arguments all of int type.
  - First two arguments are left-top corner of the rectangle, and last two arguments are right bottom corner of the rectangle.

ractangle(left, top,right, bottom);

- ractangle(100,100,200, 200);
  - Output:



```
bar(left, top,right, bottom);
```

bar(left, top,right, bottom);

- Explanation:
  - bar is used to draw a rectangle filled with given pattern.
  - We use function setfillstyle(Style, Color); to give any of the style/pattern to the bar
  - It takes 4 arguments all of int type.
  - First two arguments are left-top corner of the rectangle, and last two arguments are right bottom corner of the rectangle.

setfillstyle(STYLE,COLOR);

setfillstyle(STYLE,COLOR);

- Explanation:
  - The setfillstyle(STYLE,COLOR); sets the fill pattern and color.
  - Total of 13 styles are available to C-Compiler, which are as under:

EMPTY_FILL, SOLID_FILL, LINE_FILL, LTSLASH_FILL, SLASH_FILL, BKSLASH_FILL, LTBKSLASH_FILL, HATCH_FILL, XHATCH_FILL, INTERLEAVE_FILL, WIDE_DOT_FILL, CLOSE_DOT_FILL, USER_FILL.

We can specify the color of the object either by writing directly color name all in CAPITAL LETTERS like RED, GREEN, or by writing a corresponding equillent number of the color, like 0 for BLACK, 1 for BLUE and so on.

Similarly the fill pattern can also be replaced by their corresponding numbers, ie 0 for EMPTY_FILL, 1 for SOLID_FILL , and so on.

Hence setfillstyle(SOLID_FILL, BLUE); is equall to setfillstyle(1, 1);
Both will yield the same result.

setfillstyle(STYLE,COLOR);

Example1:

```
setfillstyle(EMPTY_FILL,BLUE);
bar(20,20,100,200);
```

Example2:

```
setfillstyle(SOLID_FILL,BLUE);
bar(20,20,100,200);
```

Similarly try other styles and colors and Enjoy! ☺

ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

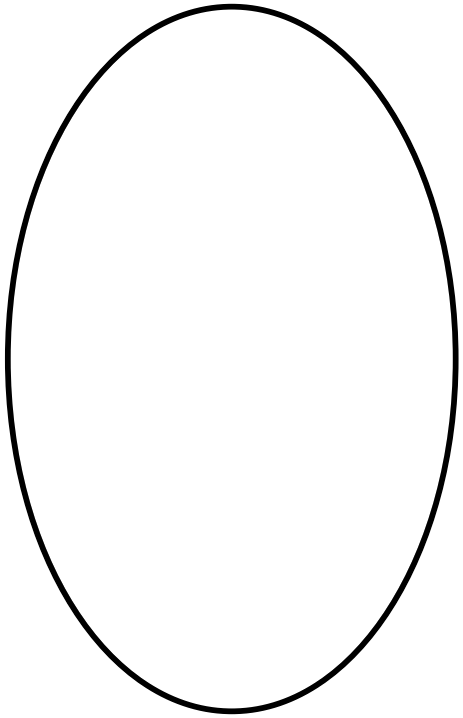
ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

- Explanation:
  - Ellipse is used to draw an elliptical arc.
  - Ellipse takes 6 arguments, all of the int type.
  - First two arguments define the center of the ellipse to place on the screen.(ie x and y co-ordinates)
  - Third and Fourth arguments are starting and ending angles of the ellipse.
  - Fifth argument is the radius of the ellipse in x-axis, and sixth argument is the radius of the ellipse in y-axis.

ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

- Example1:

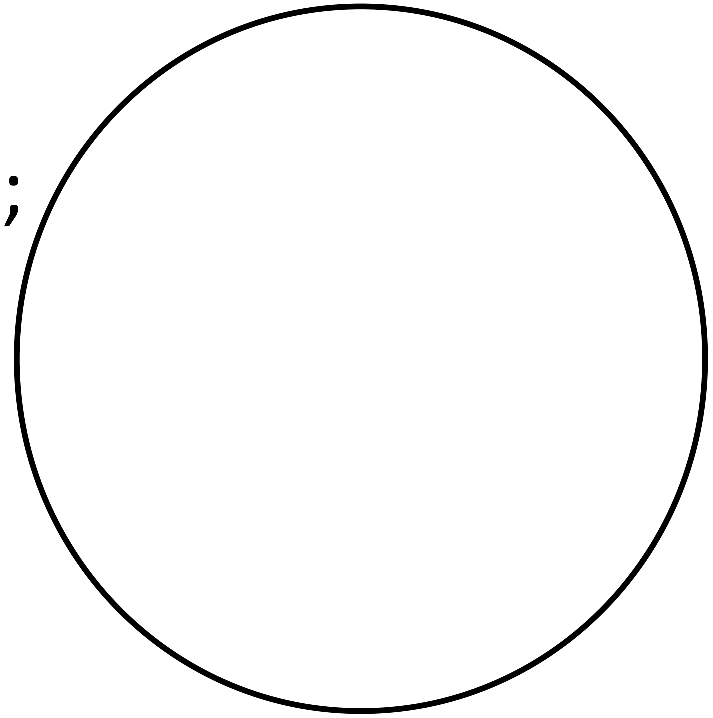
ellipse(320,240,0,360,50,100);



ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

- Example2:

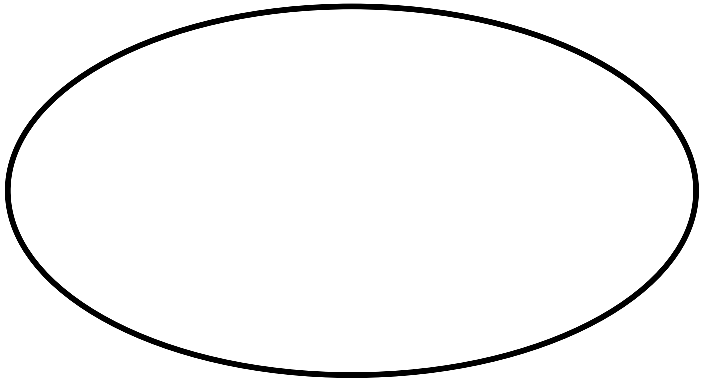
ellipse(320,240,0,360,100,100);



ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

- Example3:

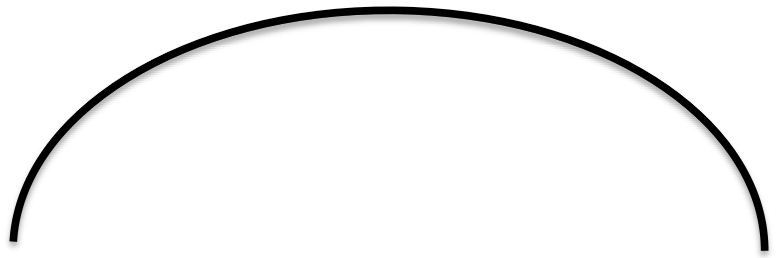
ellipse(320,240,0,360,100,50);



ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

- Example4:

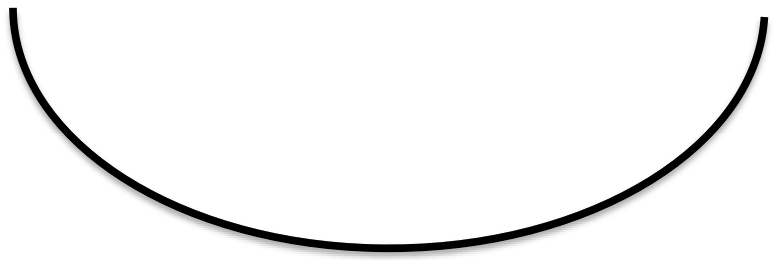
```
ellipse(320,240,0,180,100,50);
```



ellipse(midx, midy,starting-angle,ending-angle,radius-x, radius-y);

- Example5:

ellipse(320,240,180,0,100,50);



```
fillellipse(midx, midy,radius-x, radius-y);
```

fillellipse(midx, midy,radius-x, radius-y);

- Explanation:
  - Fillellipse is used to draw and fill and ellipse with given style and color.
  - Fillellipse takes 4 arguments, all of the int type.

  - First two arguments define the center of the ellipse to place on the screen.(ie x and y co-ordinates)

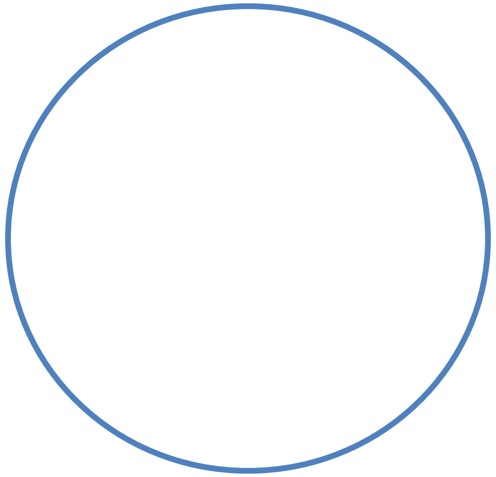
  - Third argument is the radius of the ellipse in x-axis, and fourth argument is the radius of the ellipse in y-axis.

fillellipse(midx, midy,radius-x, radius-y);

Example1:

setfillstyle(EMPTY_FILL,BLUE);

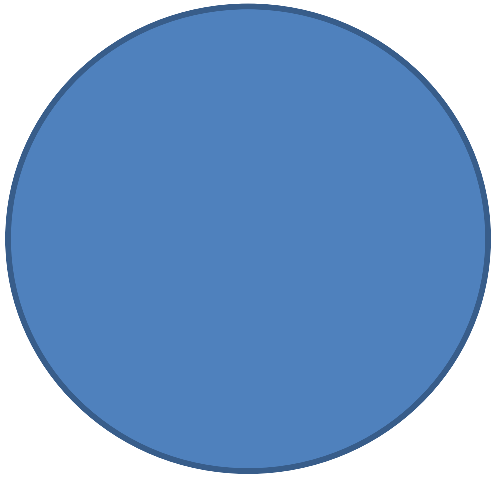
Fillelipse(320,240,100,100)



Example2:

setfillstyle(SOLID_FILL,BLUE);

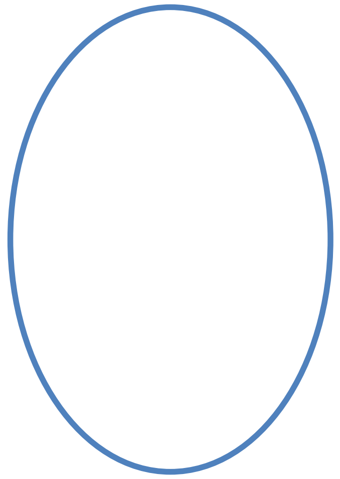
Bar(320,240,100,100);



fillellipse(midx, midy,radius-x, radius-y);

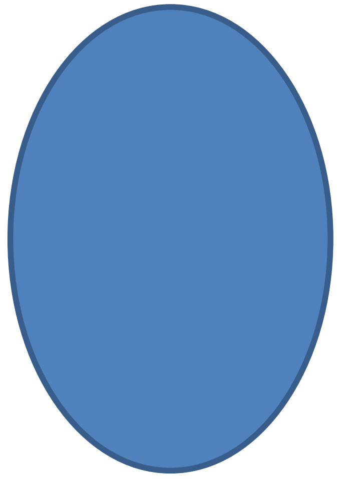
Example3:

```
setfillstyle(EMPTY_FILL,BLUE);
Fillelipse(320,240,50,100)
```



Example4:

```
setfillstyle(SOLID_FILL,BLUE);
Bar(320,240,50,100);
```

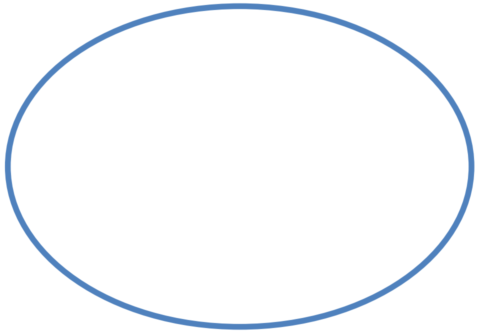


fillellipse(midx, midy,radius-x, radius-y);

Example5:

setfillstyle(EMPTY_FILL,BLUE);

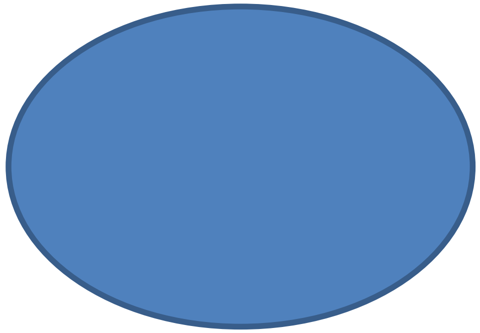
Fillelipse(320,240,100,50)



Example6:

setfillstyle(SOLID_FILL,BLUE);

Bar(320,240,100,50);



Setting font/text Style and Size

```
settextstyle(Style, Horizontal/Vertical, Size);

outtextxy(x-position, y-position, Text)
```

settextstyle(Font, Direction, Size);
outtextxy(x-position, y-position, Text)

- Explanation:
  - Settextstyle sets the Font Style, Direction and size of the Text written in outtextxy function.
  - Available Font Styles in C Compiler are as under:

Font Style	Value	Meaning
DEFAULT_FONT	0	8x8 bit-mapped font
TRIPLEX_FONT	1	Stroked triplex font
SMALL_FONT	2	Stroked small font
SANS_SERIF_FONT	3	Stroked sans-serif font
GOTHIC_FONT	4	Stroked gothic font

settextstyle(Font, Direction, Size); outtextxy(x-position, y-position, Text)

– There are two available Directions for settextstyle.

Name	Value	Direction
HORIZ_DIR	0	Left to Right
VERT_DIR	1	Bottom to Top

settextstyle(Font, Direction, Size); outtextxy(x-position, y-position, Text)

- Example1:

```
settextstyle(DEFAULT_FONT , HORIZ_DIR,1);
outtextxy(320,240,”Hello World”);
```

The above code will generate an output “Hello World”, written in default font/simple font, having horizontal direction, and text size of 1.

It will be displayed in the mid of the output window.

settextstyle(Font, Direction, Size); outtextxy(x-position, y-position, Text)

- NOTE:
  - You can also use number value instead of Font name and direction.
  - Hence:

settextstyle(DEFAULT_FONT , HORIZ_DIR,1);

is same as:

settextstyle(0 , 0,1);