

# COMPUTER PROGRAMMING

## Graphics

DR. USMAN AKMAL



DEPARTMENT OF CIVIL ENGINEERING  
UNIVERSITY OF ENGINEERING AND TECHNOLOGY,  
LAHORE

## 1 LEARNING ABOUT SCREEN

Resolution refers to the number of lines and columns on computer screen. Default screen has 80 columns and 25 lines. The intersection of a row and a column represents a picture element, or a pixel. The more rows and columns on screen result in more intersection, size of pixel will be smaller and screen resolution will increase.

All graphics on computers are composed of many pixels. Lines, boxes and circles really are just groups of pixels turned on while others are turned off.

The upper-left pixel is called the home location; this pixel is located at column 0, row 0 and, therefore, has the designation (0, 0). A pixel located at (34, 50), for example, is an intersection of graphics column 34 and graphics row 50.

To display graphics, screen must be in graphics mode. If text mode is changed to graphics mode (using screen command), program erases text on-screen in making the mode change.

Following is the syntax of the SCREEN command;

```
SCREEN mode [, colorswitch]
```

<b>Screen Mode</b>	<b>Description</b>
0	QBasic default for text resolution of 80×25. Supports up to 16 colors (0 – 15)
1	320 × 200 graphics resolution. Also supports 40 × 25 text mode and 4 of 16 colors.
2	640 × 200 graphics resolution. Also supports 80 × 25 text mode and 2 of 16 colors (Black and White).
7	320 × 200 graphics resolution. Also supports 40 × 25 text mode and 16 colors.
8	640 × 200 graphics resolution. Also supports 80 × 25 text mode and 16 colors.
9	640 × 350 graphics resolution. Also supports 80 × 25 text mode and 16 colors.
10	640 × 350 graphics resolution. Also supports 80 × 25 text mode and 2 colors.
11	640 × 480 graphics resolution. Also supports 80 × 25 text mode and 2 colors.
12	640 × 480 graphics resolution. Also supports 80 × 30 text mode and 16 colors.
13	320 × 200 graphics resolution. Also supports 40 × 25 text mode and 256 colors.

## SCREEN 12

This sets up the graphics hardware and selects mode 12. Selecting mode 12 is like selecting a sheet of graph paper. When you select mode 12 you get:

A range of X from 0 to 639.

A range of Y from 0 to 479.

16 possible colors for pixels.

With 640 columns and 480 rows you have 640 times 480 squares on the graph paper. Or, there are  $640 * 480 = 307,200$  pixels on the graphics screen. If you had actual graph paper with this many columns and rows, the squares on the paper would be very small.

The PSET statement stands for Pixel Set. A pixel is a dot on a computer graphics screen. Pixel stands for picture element. "Set" means to "turn on," so this statement turns on one dot on the screen. All computer graphics pictures are made of thousands of pixels. The PSET statement looks like this:

PSET (column, row) or (x, y)

Column and row can be numbers or variables. The part of the statement (column, row) says where on the graphics screen to set the pixel.

```
'SET A PIXEL AT (150, 32)
```

```
'
```

```
SCREEN 12
```

```
PSET (150, 32)
```

```
END
```

```
'NEW PROGRAM BASED ON THE OLD
```

```
'
```

```
SCREEN 12
```

```
PSET (150, 32) ' set X = 150 Y = 32
```

```
PSET ( 50, 30) ' set X = 50 Y = 30
```

```
PSET ( 75, 42)
```

```
PSET ( 25, 72)
```

```
PSET ( 10, 90)
```

```
PSET ( 0, 0)
```

```
PSET ( 13, 5)
```

```
END
```

The new program sets seven different pixels. Usually you do not set pixels one by one like this; better ways to draw pictures will be described later.

Here is a program that sets six pixels; the first three to green (as in the previous program) and the last three to red:

```
' Setting pixels to color 2 (Green)
```

```
' and color 4 (Red)
```

```
'
```

```
SCREEN 12
```

```
'
```

```
' Change the pen color to Green
```

```
'
```

```
COLOR 2
```

```
PSET ( 1, 3 ) ' set column 1 row 3
```

```
PSET ( 3, 3 ) ' set column 3 row 3
```

```
PSET ( 5, 3 ) ' set column 5 row 3
```

```
PSET ( 7, 3 ) ' set column 7 row 3
```

```
'
```

```
' Change the pen color to Red
```

```
'
```

```
COLOR 4
```

```
PSET ( 1, 5 ) ' set column 1 row 5
```

```
PSET ( 3, 5 ) ' set column 3 row 5
```

```
PSET ( 5, 5 ) ' set column 5 row 5
```

```
PSET ( 7, 5 ) ' set column 7 row 5
```

## Lines

In computer graphics, a line is made by setting many adjacent pixels to the same color. You could draw a line by using the PSET statement many times. But there is an easier way. To draw a line of pixels all of the same color use the LINE statement:

```
LINE (startX, startY)-(endX, endY)
```

This statement:

- Uses the current pen color.

- Sets all the pixels along a line starting at (startX, startY) and ending at (endX, endY)

- If a pixel along the line has already been set, it is set to the current pen color.

Here is a program that does the same as the previous program, except using a LINE statement:

```
' Setting 10 pixels in row 3  
' to color 4 (Red)  
'  
SCREEN 12  
COLOR 4  
LINE (1, 3) - (10, 3) ' draw a line from (1,3) to (10,3)  
END
```



## LINE COMMAND SYNTAX

To draw a line and format it:

```
LINE [[STEP] (x1, y1)] - [STEP] (x2, y2)  
[[, color] [, [B][F]] [, Style]]
```

To draw a line only:

```
LINE [[STEP] (x1, y1)] - [STEP] (x2, y2)
```

To draw a line and to give it a color

```
LINE [[STEP] (x1, y1) ] - [STEP] (x2, y2) [[, color]  
[, [B][F]] [, Style]]
```

**Screen 12**

**Line (100, 50) – (160, 50) , 3**



**Draw a box of width 60 and height 40**



## To draw a box using PSET and LINE commands

```
SCREEN
```

```
PSET (100,50)
```

```
LINE - STEP (60, 0)    `AB line
```

```
LINE - STEP (0, 40)   `BC line
```

```
LINE - STEP (-60, 0)  `CD line
```

```
LINE - STEP (0, -40)  `DA line
```

To draw a filled box and give it a color

```
LINE [[STEP] (x1, y1) ] - [STEP] (x2, y2) [[, color]  
[, [B][F]] [, style]]
```

**Screen 12**

**Line (100, 50) – (160, 90) , 3, BF**

To draw a filled box and give it a style code

```
LINE [[STEP] (x1, y1)] - [[STEP] (x2, y2) [[, color]  
[, [B][F]] [, Style]]
```

**, &HF0F0 – Hexadecimal code**

**Style is represented in 16 bit pattern. Decide how many will be ON and how many will be OFF for the pattern.**

, &HF0F0 – Hexadecimal code

Style is represented in 16 bit pattern. Decide how many will be ON and how many will be OFF for the pattern.

True Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

1111	0000	1111	0000
15 (F)	0	15(F)	0

If you do not give any code, a solid line will be drawn

SCREEN 7

LOCATE 2: PRINT "DASHED LINES"

LINE (20, 30) - (300, 30), , , &HAAAA '1010101010101010

LINE (20, 45) - (300, 45), , , &HCCCC '1100110011001100

LINE (20, 60) - (300, 60), , , &HFF00 '1111111100000000

LINE (20, 75) - (300, 75), , , &HFFF0 '11111111111110000

LOCATE 14: PRINT "CENTER LINES"

LINE (20, 120) - (300, 120), , , &HF3FC '1111001111111100

LINE (20, 135) - (300, 135), , , &HE3F8 '1110001111111000

LINE (20, 150) - (300, 150), , , &HFCCC '1111110011001100

LINE (20, 151) - (300, 151), , , &HFCCC '1111110011001100





END OF LECTURE