# Lab 1: Introduction to graphics primitive and graphics drivers

1. Software requirement: Turbo C / C++

**BASIC GRAPHICS FUNCTION**

1. **INITGRAPH**

• Initializes the graphics system.

**Declaration**

• Void far initgraph(int far \*graphdriver)

**Remarks**

• To start the graphic system, you must first call initgraph.

• Initgraph initializes the graphic system by loading a graphics driver from disk (or validating a registered driver) then putting the system into graphics mode.

• Initgraph also resets all graphics settings (color, palette, current position, viewport, etc) to their defaults then resets graph.

1. **GETPIXEL, PUTPIXEL**

• Getpixel gets the color of a specified pixel.

• Putpixel places a pixel at a specified point.

**Decleration**

• Unsigned far getpixel(int x, int y)

• Void far putpixel(int x, int y, int color)

**Remarks**

• Getpixel gets the color of the pixel located at (x,y);

• Putpixel plots a point in the color defined at (x, y)

**Return value**

• Getpixel returns the color of the given pixel.

• Putpixel does not return

1. **CLOSE GRAPH**

• Shuts down the graphic system.

**Declaration**

• Void far closegraph(void);

**Remarks**

• Close graph deallocates all memory allocated by the graphic system.

• It then restores the screen to the mode it was in before you called initgraph.

**Return value**

• None.

1. **ARC, CIRCLE, PIESLICE**

• arc draws a circular arc.

• Circle draws a circle

• Pieslice draws and fills a circular pieslice

**Declaration**

• Void far arc(int x, int y, int stangle, int endangle, int radius);

• Void far circle(int x, int y, int radius);

• Void far pieslice(int x, int y, int stangle, int endangle, int radius);

**Remarks**

• Arc draws a circular arc in the current drawing color

• Circle draws a circle in the current drawing color

• Pieslice draws a pieslice in the current drawing color, then fills it using the current fill pattern and fill color.

1. **ELLIPSE, FILL ELIPSE, SECTOR**

• Ellipse draws an elliptical arc.

• Fill ellipse draws and fills ellipse.

• Sector draws and fills an elliptical pie slice.

**Declaration**

• Void far ellipse (int x, int y, int stangle, int endangle, int xradius, int yradius)

• Void far fill ellipse (int x, int y, int xradius, int yradius)

• Void farsectoe(int x, int y, int stangle, int endangle, int xradius, int yradius)

**Remarks**

• Ellipse draws an elliptical arc in the current drawing color.

• Fill ellipse draws an elliptical arc in the current drawing color and then fills it with fill color and fill pattern.

• Sector draws an elliptical pie slice in the current drawing color and then fills it using the pattern and color defined by setfill style or setfill pattern.

1. **FLOODFILL**

• Flood-fills a bounded region.

**Declaration**

• Void far floodfill(int x, int y, int border)

**Remarks**

• Floodfills an enclosed area on bitmap device.

• The area bounded by the color border is flooded with the current fill pattern and fill color.

• (x,y) is a “seed point”

¬ If the seed is within an enclosed area, the inside will be filled.

¬ If the seed is outside the enclosed area, the exterior will be filled.

• Use fillpoly instead of floodfill wherever possible so you can maintain code compatibility with future versions.

• Floodfill doesnot work with the IBM-8514 driver.

**Return value**

• If an error occurs while flooding a region, graph result returns „1‟.

**7) GETCOLOR, SETCOLOR**

• Getcolor returns the current drawing color.

• Setcolor returns the current drawing color.

**Declaration**

• Int far getcolor(void);

• Void far setcolor(int color)

**Remarks**

• Getcolor returns the current drawing color.

• Setcolor setsthe current drawing color to color, which can range from 0 to getmaxcolor.

• To set a drawing color with set color , you can pass either the color number or the equivalent color name.

1. **LINE, LINEREL, LINETO**

• Line draws a line between two specified pints.

• Onerel draws a line relative distance from current position (CP).

• Linrto draws a line from the current position (CP) to(x,y).

• Void far lineto(int x, int y)

**Remarks**

• Line draws a line from (x1, y1) to (x2, y2) using the current color, line style and thickness. It does not update the current position (CP).

• Linerel draws a line from the CP to a point that is relative distance (dx, dy) from the CP, then advances the CP by (dx, dy).

• Lineto draws a line from the CP to (x, y), then moves the CP to (x,y).

**Return value**

• None

1. **RECTANGLE**

• Draws a rectangle in graphics mode.

**Decleration**

• Void far rectangle (int left, int top, int right, int bottom)

**Remarks**

• It draws a rectangle in the current line style, thickness and drawing color.

• (left, top) is the upper left corner of the rectangle, and (right, bottom) is its lower right corner.

**Return value**

• None.