

# UNIT 11

# INTRODUCTION TO GRAPHICS

LH - 3HRS

PRESENTED BY: **ER. SHARAT MAHARJAN**

C PROGRAMMING

PRIME COLLEGE, NAYABAZAAR

# CONTENT (LH - 3HRS)

11.1 Introduction,

11.2 Graphical Initialization and Modes,

11.3 Graphical Functions

# 11.1 Introduction

- There are two modes of output device:
  - ✓ Text mode
  - ✓ Graphics mode
- In C graphics, the **functions** in header file **graphics.h** are used to draw different shapes like circles, rectangles, etc, display text(any message) in a different format (different fonts and colors).

## Some terms used in Graphics Programming

- **Pixel** is a single point in a graphic image.
- **Resolution** is the total number of pixels setup for a screen.
- **Video Adapters** are drivers for display.

# 11.2 Graphical Initialization and Modes

- Initializing Graphics Mode:

We use `initgraph()` function as:

```
initgraph(&graphics_driver, &graphics_mode, "path_to_driver");
```

- Closing Graphics Mode:

We use `closegraph()` function as:

```
closegraph();
```

- Example:

```
int gd=DETECT,gm;//auto detection of graphics driver VGA,CGA,EGA  
initgraph(&gd, &gm, "C:\\TC\\BGI");//mode-resolution, color available  
closegraph();
```

## Result of Graphics Operations:

- The library function **graphresult()** is used to determine whether a certain graphics operation succeeded or not.
- The function returns 0 if no error occurs.

# 11.3 Graphical Functions

## 1. Plotting and getting points

- putpixel(): Plots a point with a specified color

**Syntax: putpixel(int x, int y, int color);**

- getpixel(): Gets color of specified pixel

**Syntax: integer\_variable=getpixel(int x, int y);**

## 2. Changing drawing/foreground and background color

- setcolor(): It changes current drawing/foreground color.

**Syntax: setcolor(int color);**

- setbkcolor(): It changes the background color.

**Syntax: setbkcolor(int color);**

### 3. Drawing lines

- line(): It draws a line from point having co-ordinate x1, y1 to x2, y2.

**Syntax: line(int x1, int y1, int x2, int y2);**

**LAB 1: WAP to draw a line passing from a point (0,50) to another point (200,300).**

```
#include<stdio.h>
```

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

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

```
#include<stdlib.h>
```

```
int main(){
```

```
    int gd=DETECT,gm;
```

```
    initgraph(&gd,&gm,"C:\\TC\\BGI");
```

```
    if(graphresult()!=0){
```

```
        printf("Initialization failed.");
```

```
    }
```

```
    setcolor(GREEN);
```

```
    line(0,50,200,300);
```

```
    getch();
```

```
    closegraph();
```

```
    return 0;
```

```
}
```



#### 4. Drawing Circles

- **circle()**: It draws a circle having center point (x,y) and radius r.

**Syntax: circle(int x, int y, int r);**

**LAB 2: WAP to draw two concentric circles with center (200,200) and radii 50 and 75.**

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
int main(){
    int gd=DETECT, gm;
    initgraph(&gd,&gm,"C:\\\\TC\\\\bgi");
    if(graphresult()!=0){
        printf("Initialization failed.");
        exit(0);
    }
    circle(200,200,50);
    circle(200,200,75);
    getch();
    closegraph();
    return 0;
}
```

## 5. Drawing ellipses

- **ellipse()**: It draws an ellipse.

**Syntax: ellipse(int x, int y, int startAngle, int endAngle, int xRadius, int yRadius);**

## 6. Drawing circular arcs

- **arc()**: It draws a circular arc.

**Syntax: arc(int x, int y, int startAngle, int endAngle, int radius);**

## 7. Drawing rectangles

- **rectangle()**: It draws rectangle from two end points of a diagonal of the rectangle.

**Syntax: rectangle(int x1, int y1, int x2, int y2);**

## 8. Drawing and filling a polygon

- **drawpoly()**: It draws the outline of a polygon.
- **fillpoly()**: It draws and fills a polygon.

**Syntax: drawpoly(int numberOfPoints, int points[]);**  
**fillpoly(int numberOfPoints, int points[]);**

## 9. Displaying text in graphics mode

- **outtext()**: It displays the string at the current position.

**Syntax: outtext(string text);**

- **outtextxy()**: It displays the string at point(x,y).

**Syntax: outtextxy(int x, int y, string text);**

- **settextstyle()**: It changes font, size and direction of characters.

**Syntax: settextstyle(int font, int direction, int size);**

**THANK YOU FOR YOUR ATTENTION**