

Graphics Programming

16.1 Introduction

One of the most important features of Java is its ability to draw graphics. We can write Java applets that can draw lines, figures of different shapes, images, and text in different fonts, styles and colours.

The AWT supports number of graphics methods. All graphics are drawn relative to a window. The window can be an applet window or a stand-alone application window.

Every applet has its own area on the screen known as **canvas**, where it creates display. Java coordinates system has the origin (0, 0) in the upper-left corner. Positive x values are to the right and +ve y values to the bottom as shown in figure 16.1.

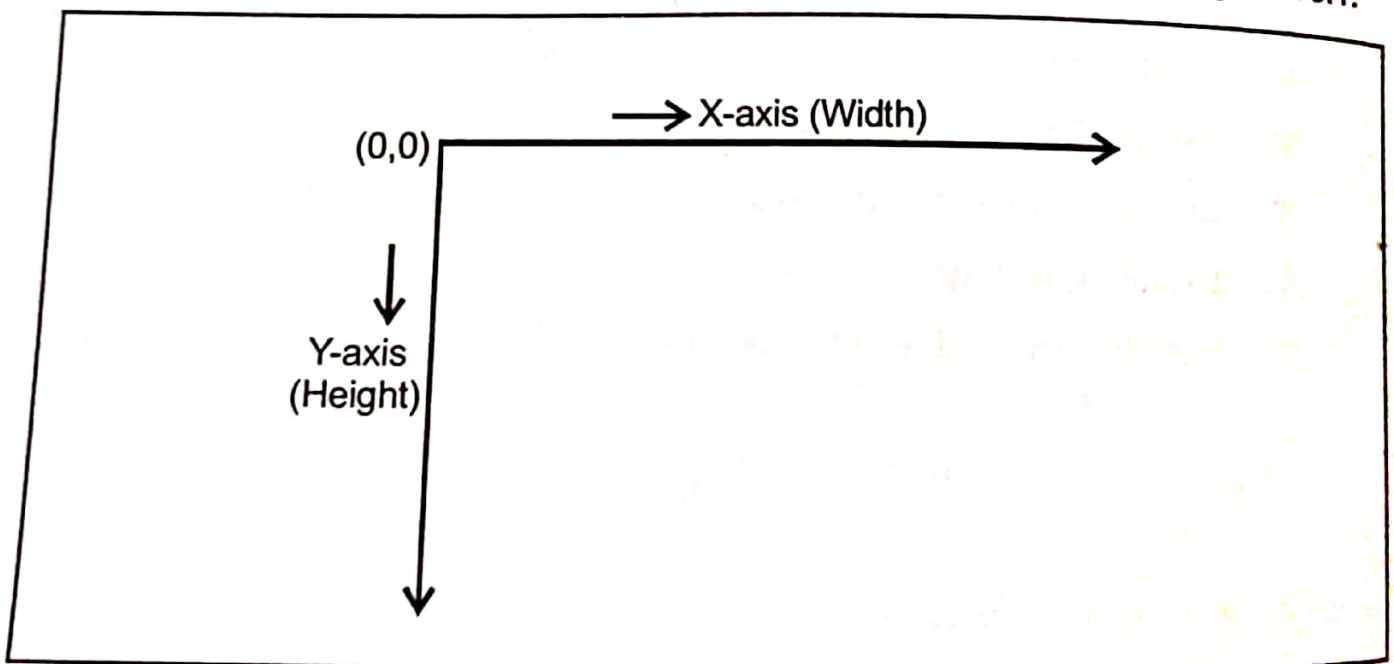


FIGURE 16.1

java.awt.Graphics is an abstract class that defines a number of graphical operations. Its method can be used for rendering images and text, drawing and filling shapes, clipping graphical operations, and much more.

16.2 Working With Color

The AWT color system allows you to specify any color you want. Java colors are represented by the combinations of the numeric intensities of the primary colors (red, green and blue). This color system is known as RGB (Red Green Blue). We can create our own colors, using following color constructor.

Color(int red, int green, int blue) : Constructor used to create color.

This constructor takes three integers that specify the color as a mix of red, green and blue. These values must be between 0 and 255.

For example :

```
Color c=new Color(255, 0, 0); //Red
```

Some commonly used methods are :

Methods	Description
setColor(Color c)	To set the color
getColor()	To obtain the current color
getRed()	To obtain the red component in integer (from 0 to 255) of the current color
getGreen()	To obtain green component (from 0 to 255) of the current color
getBlue()	To obtain blue component (from 0 to 255) of the current color
getRGB()	To get a packed, RGB representation of a color
setBackground()	To set the background color
setForeground()	To set the Foreground color. By default, all graphics objects are drawn in the current foreground color

Example

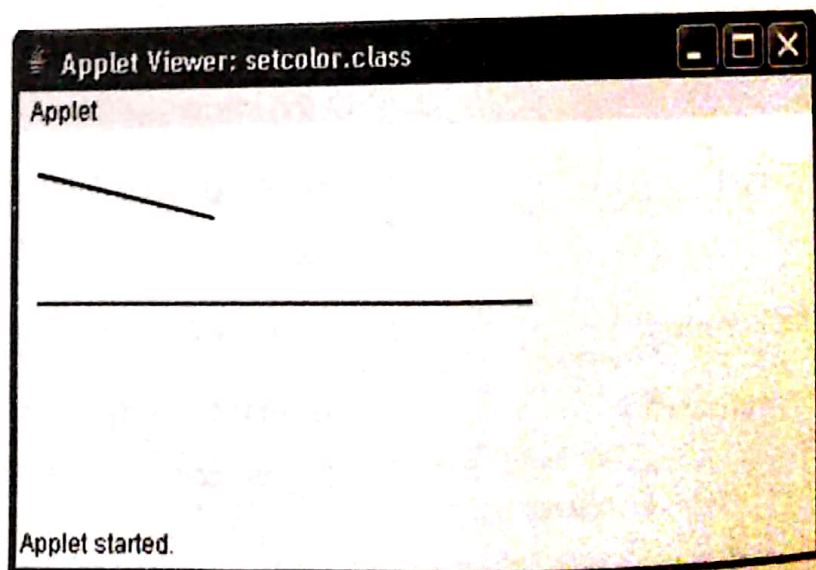
//Code of setcolor.java

```
import java.awt.*;
import java.applet.*;
public class setcolor extends Applet
{
    public void paint(Graphics g)
    {
        Color c1 = new Color (255,0,0); //Red
        Color c2 = new Color (192,192,192); //Light Gray
        g.setColor(c1);
        g.drawLine(10,20,100,40);
        g.setColor(c2);
        g.drawLine(10,80,250,80);
    }
}
```

//Code of setcolor.html

```
<HTML>
<BODY>
    <APPLET CODE = "setcolor.class"
        Width = 400
        Height = 180>
    </APPLET>
</BODY>
</HTML>
```

Output :



16.3 Working With Fonts

We can create a new font by using the following constructor :

Font(String fname, int style, int size);

Where, **fname** is the name of font.
Style is one or more of three constants, namely, Font.BOLD, Font.ITALIC and Font.PLAIN. We can use more than one style by using OR operation like FONT.BOLD | FONT.ITALIC
Size is size of font

Commonly methods are :

Methods	Description
setFont(Font f)	To set the font
getFont()	To get the font
getName()	To get name of the invoking font
getSize()	To get size of the invoking font
getStyle()	To get style values of the invoking font
isBold()	Returns true if the font includes the BOLD style value.
isItalic()	Returns true if the font includes the ITALIC style value.
isPlain()	Returns true if the font includes the PLAIN style value.

Example

//Code of setfont.java

```
import java.awt.*;  
import java.applet.*;  
public class setfont extends Applet
```



```

{
    Font f;
    public void init()
    {
        f=new Font("Arial", Font.BOLD, 18);
    }

    public void paint(Graphics g)
    {
        Color c1 = new Color (0,0,255); //Blue
        g.setColor(c1);
        g.setFont(f);
        g.drawString("Aarti Book Company",80,120);
    }
}

```

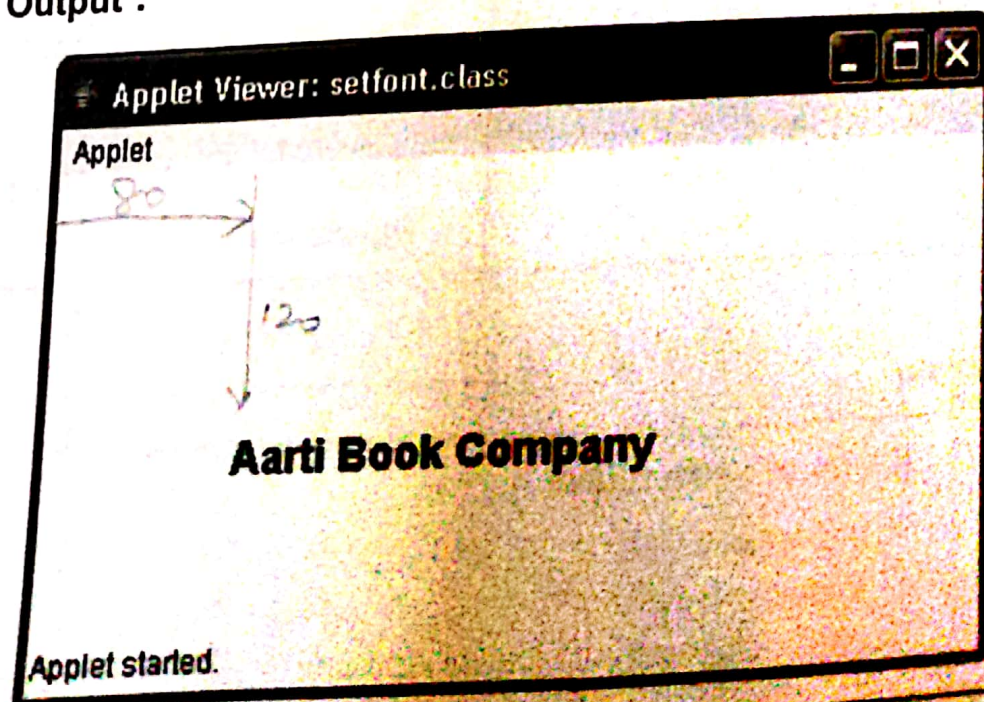
//Code of setfont.html

```

<HTML>
<BODY>
    <APPLET CODE = "setfont.class"
        Width = 400
        Height = 180>
    </APPLET>
</BODY>
</HTML>

```

Output :



16.4 Drawing Lines

Lines are drawn by means of the `drawLine()` methods. Syntax is

```
drawLine(int x1, int y1, int x2, int y2);
```

Where,

x1 : the first point's x coordinate.

y1 : the first point's y coordinate.

x2 : the second point's x coordinate.

y2 : the second point's y coordinate.

Note that, The AWT is not capable of drawing lines of variable thickness.

Example

//Code of demoline.java

```
import java.awt.*;
import java.applet.*;
public class demoline extends Applet
{
    public void paint(Graphics g)
    {
        g.drawLine(0,0,180,240);
        g.drawLine(0,50,82,190);
        g.drawLine(80,50,180,50);
        g.drawLine(100,80,220,80);
    }
}
```

//Code of demoline.html

```
<HTML>
<BODY>
    <APPLET CODE = "demoline.class"
        Width = 400
        Height = 180>
    </APPLET>
</BODY>
</HTML>
```

Output :

