



Sri  
**SAI RAM**  
ENGINEERING COLLEGE  
INSTITUTE OF TECHNOLOGY  
West Tambaram, Chennai - 44

**Sairam**  
INSTITUTIONS



**SAIRAM**  
DIGITAL RESOURCES

## UNIT NO 5- EVENT DRIVEN PROGRAMMING

### 5.2 Working with 2D shapes - Using color, fonts, and images

YEAR	SEM

**CS8392**

**OBJECT ORIENTED PROGRAMMING**  
(Common to EEE, CSE, EIE, ICE, IT)

**COMPUTER SCIENCE & ENGINEERING**



## Working with 2D shapes

Java supports 2 dimensional, shapes, text and images using methods available in graphics 2D class.

The Graphics 2D class extends the Graphics to provide more sophisticated control over geometry, Co ordinate transformations, color management and text layout.

Graphics 2D class constructor

Graphics2D() //constructs a new Graphics2D object

This class inherits the methods from java.lang.Object. Some of the commonly used methods of Graphics2D class are as follows

## Methods

Method	Description
void draw(Shape s)	Strokes the outline of a Shape using the settings of the current Graphics2D context
void draw3DRect(int x, int y, int width, int height, boolean raised)	Draws a 3-D highlighted outline of the specified rectangle
void drawImage(BufferedImage img, BufferedImageOp op, int x, int y)	Renders a BufferedImage that is filtered with a BufferedImageOp
boolean drawImage(Image img, AffineTransform xform, ImageObserver obs)	Renders an Image applying a transform from image space into user space before drawing
void drawString(String str, float x, float y)	Renders the text specified by the string using the current text attribute state in the Graphics 2D context
void fill(Shape s)	Fills the interior of a shape using the settings of the Graphics 2D context

## Methods

Method	Description
void rotate(double theta)	Concatenates the current Graphics2D transform with rotation transform
void scale(double sx, double sy	Concatenates the current Graphics 2D transform with scaling transformation subsequent rendering is resized according to the specified scaling factors relative to the previous scaling
void setBackground(Color color)	Sets the background color for the Graphics2D context.
void setStroke(Stroke s)	Sets the Stroke for the Graphics2D context.
void shear(double shx, double shy)	Concatenates the current Graphics 2D transform with a shearing transform
Void transform(AffineTransform Tx)	Composes an AffineTransform object with the Transform in this Graphics2D according to the rulelast-specified-first-applied

## Example

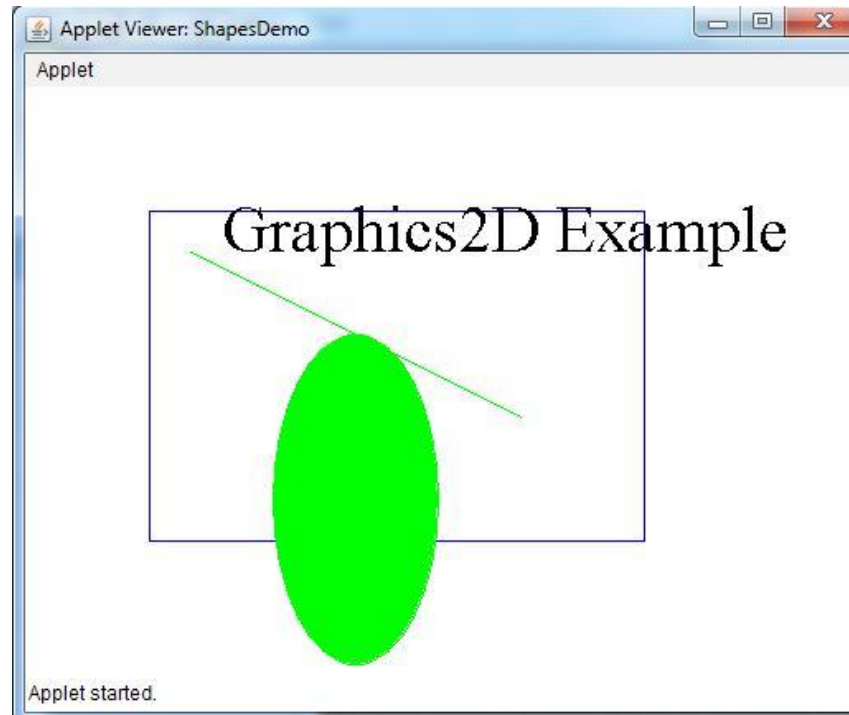
### ***Example:***

```
import java.awt.*;  
  
import java.applet.*;  
  
/*  
<applet code="ShapesDemo" width=350 height=300>  
</applet>  
  
*/  
  
public class ShapesDemo extends Applet {  
  
    public void init() {}  
  
    public void paint(Graphics g) {  
  
        Graphics2D g2d = (Graphics2D)g;
```

## Example

```
g2d.setColor(Color.blue);  
  
g2d.drawRect(75,75,300,200);  
  
Font exFont = new Font("TimesRoman",Font.PLAIN,40);  
  
g2d.setFont(exFont);  
  
g2d.setColor(Color.black);  
  
g2d.drawString("Graphics2D Example",120,100);  
  
g2d.setColor(Color.green);  
  
g2d.drawLine(100,100,300,200);  
  
g2d.drawOval(150,150,100,200);  
  
g2d.fillOval(150,150,100,200);  
  
}  
  
}
```

## Output



## Colors

To support different colors Java package comes with the Color class. The Color class states colors in the default is RGB color space or colors in arbitrary color spaces identified by a ColorSpace.

Color class static color variables available are:

Color.orange	Color.black
Color.blue	Color.cyan
Color.darkGray	Color.grayColor.green
Color.yellow	Color.lightGray
Color.magenta	Color.pink
Color.red	Color.white

Color class constructor

Color(float r, float g, float b) – create color with specified red, green, and blue values in the range (0.0 - 1.0)





## Methods in Colors

Color(int r, int g, int b)- create color with the specified red, green, and blue values in the range (0 - 255).

Color class constructor

Method	Description
int getRed()	Returns the red component in the range 0-255 in the default RGB space
int getGreen()	Returns the green component in the range 0-255 in the default RGB space.
int getBlue()	Returns the blue component in the range 0-255 in the default RGB space.
Color getHSBColor(float h, float s, float b)	Creates a Color object based on the specified values for the HSB color model.



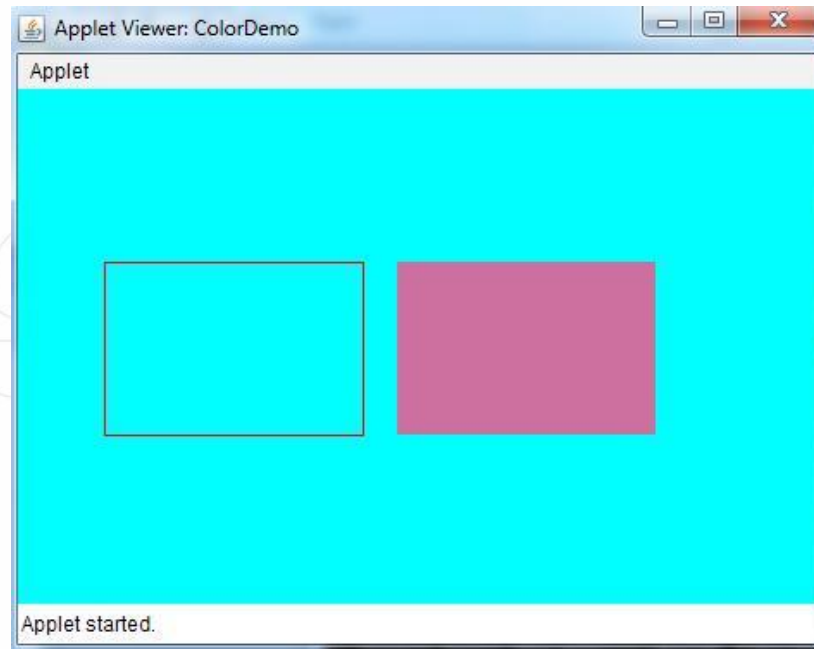
## Example Program

The current graphics color can be changed using setColor() method defined in Graphics class

```
import java.awt.*;
import java.applet.*;
/*
<applet code="ColorDemo" width=350 height=300>
</applet>
*/
public class ColorDemo extends Applet {
    public void init() {
        setBackground(Color.CYAN);
    }
    public void paint(Graphics g) {
        g.setColor(Color.red)
        g.drawRect(50, 100, 150, 100); // rectangle outline is red color
        Color clr = new Color(200, 100, 150);
        g.setColor(clr);
        g.fillRect(220,100, 150, 100);
    }
}
```



output



## Fonts

### Fonts in Java

The Font class states fonts, which are used to render text in a visible way.

### Font class constructor

Font(Font font)

//Creates a new Font from the specified font.

Font(String name, int style, int size)

//Creates new Font from the specified name, style  
and point size

## Fonts class

*Font variables available in Font class are:*

Font.BOLD	Font. SANS_SERIF
Font.ITALIC	Font. CENTER_BASELINE
Font. PLAIN	Font. DIALOG
Font. MONOSPACED	Font. SERIF
Font. TRUETYPE_FONT	Font. TYPE1_FONT
int size	int style
float pointSize	String name

## Methods in Fonts

Method	Description
String getFamily()	Returns the family name of this Font.
int getStyle()	Returns the style of this Font.
boolean isBold()	Indicates whether or not this Font object's style is BOLD
boolean isItalic()	Indicates whether or not this Font object's style is ITALIC.
boolean isPlain()	Indicates whether or not this Font object's style is PLAIN.
static Font getFont(String nm)	Returns a Font object from the system properties list
static Font decode(String str)	Returns the Font that the str argument describes
String toString()	Converts this Font object to String representation

## Fonts

**Example:**

```
import java.applet.Applet;

import java.awt.*;

import java.awt.event.*;

/* <APPLET CODE ="FontDemo.class" WIDTH=300 HEIGHT=200>
</APPLET> */ public class FontDemo extends java.applet.Applet

{
    Font f;
    String m;
    public void init()
    {
        f=new Font("Arial",Font.ITALIC,20);
        m="Welcome to Java";
    }
}
```

## Fonts

```
setFont(f);  
}  
public void paint(Graphics g)  
{  
    Color c=new Color(100,100,255);  
    g.setColor(c);  
    g.drawString(m,4,20);  
    Font plainFont = new Font("Serif", Font.PLAIN, 24);  
    g.setFont(plainFont);  
    g.drawString("Font in PLAIN", 50, 70);  
    Font italicFont = new Font("Serif", Font.ITALIC, 24); g.setFont(italicFont);  
    g.drawString("Font in ITALIC", 50, 120);  
    Font boldFont = new Font("Serif", Font.BOLD, 24);  
    g.setFont(boldFont);
```



## Fonts

```
g.drawString("Font in BOLD", 50, 170);  
Font boldItalicFont = new Font("Serif", Font.BOLD+Font.ITALIC, 24);  
g.setFont(boldItalicFont);  
g.drawString("Font in BOLD ITALIC", 50, 220);  
}  
}
```

**Sample Output:**



## Images in java

Image control is super class for all image classes representing graphical images.

### Image class constructor

`Image()` // create an Image object

The `java.applet.Applet` class provides following methods to access image.

1. `getImage()` method that returns the object of Image. Its syntax is as follows.  
`public Image getImage(URL u, String imag ){ }`

2. `getDocumentBase()` method returns the URL of the document in which applet is embedded.

`public URL getDocumentBase (){}`

3. `getCodeBase()` method returns the base URL. `public URL getCodeBase()`

## Example

```
import java.applet.Applet;

import java.awt.*;

import java.awt.event.*;

import java.net.URL;

/* <APPLET CODE ="ImageDemo.class" WIDTH=300 HEIGHT=200> </APPLET> */

public class ImageDemo extends java.applet.Applet
{
    Image img;
    public void init()
    {

    }

    public void paint(Graphics g)
    {
```

## Example

```
URL url1 = getCodeBase();  
img = getImage(url1,"java.jpg");  
g.drawImage(img, 60, 120, this);  
}  
}
```

Sample output



## Methods in image

Method	Description
Graphics getGraphics()	Creates a graphics context for drawing to an off-screen image
int getHeight(ImageObserver observer)	Determines the height of the image.
Image getScaledInstance(int width, int height, int hints)	Creates a scaled version of this image
ImageProducer getSource()	Gets the object that produces the pixels for the image.
int getWidth(ImageObserver observer)	Determines the width of the image

## Video Link

<https://www.youtube.com/watch?v=M-F7z1xWS6o>

*Sairam*