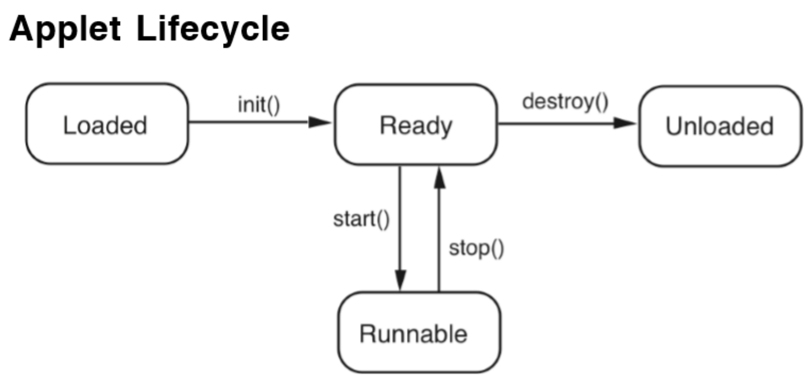
Introduction to Applet

Applets are small Internet-based program written in Java, a programming language for the Web and can be downloaded by any computer. The applet is also capable of running in HTML. The applet is usually embedded in an HTML page on a Web site and can be executed from within a browser. After an applet arrives on the client, it has limited access to resources so that it can produce a graphical user interface and run complex computations without introducing the risks of viruses or data security breaching.
This applet begins with two import statements. The first import statement is for the Abstract Window Toolkit (AWT) classes. Applets interact with the user through the AWT and not through the console-based I/O classes. An applet must be a subclass of the java.applet.Applet class. The Applet class provides the standard interface between the applet and the browser environment. Swing provides a special subclass of the Applet class called javax.swing.JApplet. The JApplet class should be used for all applets that use Swing

the lifecycle of an Applet. The applet in Java can appear in a frame of the web page, a new application window, Sun's AppletViewer, or a stand-alone tool for testing them. They were introduced in the first version of the Java language, which was introduced in the year 1995.

Life Cycle of an Applet



Four methods in the Applet class gives you the framework on which you build any serious applet –

init – This method is intended for whatever initialization is needed for your applet. It is called after the param tags inside the applet tag have been processed.

start – This method is automatically called after the browser calls the init method. It is also called whenever the user returns to the page containing the applet after having gone off to other pages.

stop – This method is automatically called when the user moves off the page on which the applet sits. It can, therefore, be called repeatedly in the same applet.

destroy – This method is only called when the browser shuts down normally. Because applets are meant to live on an HTML page, you should not normally leave resources behind after a user leaves the page that contains the applet.

paint – Invoked immediately after the start() method, and also any time the applet needs to repaint itself in the browser. The paint() method is actually inherited from the java.awt.

HTML <applet> Tag

Description

The `<applet>` tag embeds a Java applet (mini Java applications) on the page. An applet is a program written in the Java programming language that can be included in an HTML page,

Syntax

The basic syntax of the `<applet>` tag is given with:

```
HTML / XHTML:
<applet code="URL" width="length" height="length"> ... </applet>
```

The example below shows the `<applet>` tag in action.

Example

```
<applet code="Animate.class" width="400" height="300">
<param name="delay" value="30">
 <param name="time" value="120">
</applet>
```

Parameter in Applet

We can get any information from the HTML file as a parameter. For this purpose, Applet class provides a method named getParameter().
Syntax:

```
public String getParameter(String parameterName)
```

Example of using parameter in Applet:

```
import java.applet.Applet;
import java.awt.Graphics;

public class UseParam extends Applet{

public void paint(Graphics g){
String str=getParameter("msg");
g.drawString(str,50, 50);
}


}
```

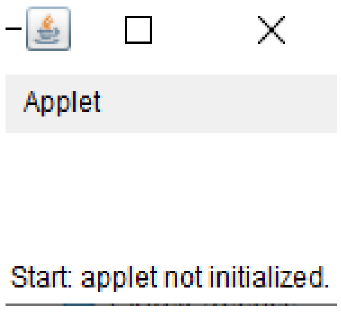
myapplet.html

```
<html>
<body>
<applet code="UseParam.class" width="300" height="300">
<param name="msg" value="Welcome to applet">
</applet>
</body>
</html>
```

Embedding Applet Tag in Java Code

By embedding applet tag inside the java program we will directly run the applet without using independent html file to test applet with Applet Viewer only for testing however such code not work on browser to use applet

```
import java.awt.*;

import java.applet.*;

class MyApplet extends  Applet
{
public void paint(Graphics g)
{
g.drawString("A Simple Applet",100,100);
}
}
/*<applet code="MyApplet.class" width=100 height=50></applet> */
```



Start: applet not initialized.

```
C:\Program Files\Java\jdk1.8.0_201\bin>javac Myapplet.java

C:\Program Files\Java\jdk1.8.0_201\bin>appletviewer Myapplet.java
```

Creating an Applet User Interface and Add Control to Applet
Most applets have a *graphical user interface* (GUI). This is a natural consequence of the fact that each applet appears, as specified by the <APPLET> tag within an HTML page that's displayed by a browser. Because the Applet class is a subclass of the AWT (Abstract Window Toolkit) Panel class, and thus participates in the AWT event and drawing model, creating an applet's GUI is just as easy as creating an application's GUI -- easier, actually, since the applet's window (the browser window) already exists.

An applet can mix and match several UI types, depending on the kind of information it needs to give or get. Some applets play *sounds*, either to give the user feedback or to provide ambiance. Applets can get configuration information from the user through *parameters* that the applet defines. Applets can get system information by reading *system properties*. To give text information to the user, an applet can use its GUI or display a *short status string* (for text that's not crucial) or display to the *standard output* or *standard error* stream (for debugging text).

Pre-Made UI Components
The AWT supplies the following UI components (the class that implements each component is listed in parentheses):

- Buttons (java.awt.Button)
- Checkboxes (java.awt.Checkbox)
- Single-line text fields (java.awt.TextField)
- Larger text display and editing areas (java.awt.TextArea)
- Labels (java.awt.Label)
- Lists (java.awt.List)
- Pop-up lists of choices (java.awt.Choice)
- Sliders and scrollbars (java.awt.Scrollbar)
- Drawing areas (java.awt.Canvas)
- Menus (java.awt.Menu, java.awt.MenuItem, java.awt.CheckboxMenuItem)

- Containers (java.awt.Panel, java.awt.Window and its subclasses)

Methods for Using UI Components in Applets
Because the Applet class inherits from the AWT Container class, it's easy to add components to applets and to use layout managers to control the components' onscreen positions. Here are some of the Container methods an applet can use:

add

    Adds the specified Component.

remove

    Removes the specified Component.

setLayout

    Sets the layout manager.

```
/*
 Change Button Font Example
 This java example shows how to change button's font using
 AWT Button class.
*/

import java.applet.Applet;
import java.awt.Button;
import java.awt.Font;

 /*
<applet code="ChangeButtonFontExample" width=200 height=200>
</applet>
*/
public class ChangeButtonFontExample extends Applet{

 public void init(){

 //create buttons
 Button button1 = new Button("Button 1");
 Button button2 = new Button("Button 2");

 /*
 * To change font of a button use
 * setFont(Font f) method.
 */

 Font myFont = new Font("Courier", Font.ITALIC,12);
 button1.setFont(myFont);

 //add buttons
 add(button1);
 add(button2);
 }
}
```

GRAPHICS CLASS

Graphics class include methods for drawing shapes , images to the screen inside your applet o Graphics class contain several inbuilt methods to create graphical interface.

Draw a simple string in Java

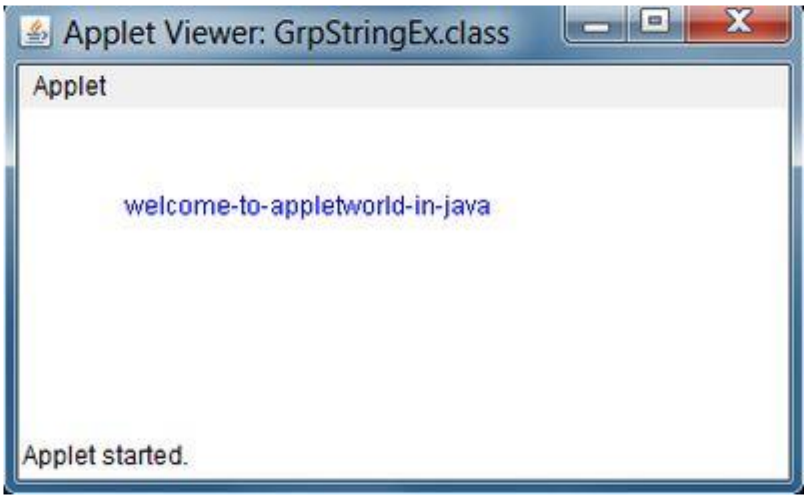
In this program, we show how to design a string in graphics programming.

```
import java.awt.*;
import java.applet.Applet;
public class GrpStringEx extends Applet {
public void paint(Graphics grp)
{
grp.setColor(Color.blue);
grp.drawString("welcome-to-appletworld-in-java", 150, 150);
}
}
/*
<applet code="GrpStringEx.class" width="400" height="400">
</applet>
*/
```

Output

To execute our code by the appletviewer tool, write in the command prompt:

```
javac GrpStringEx.java
appletviewer GrpStringEx.java
```

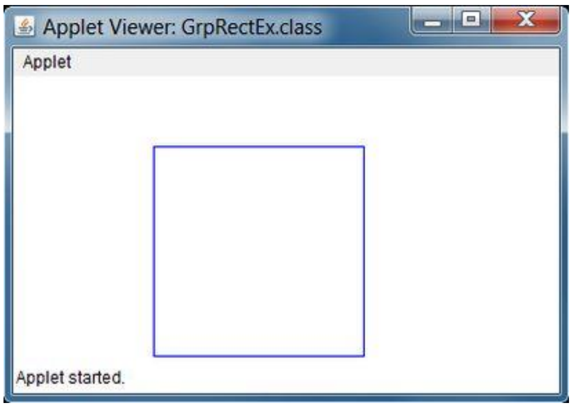


```
welcome-to-appletworld-in-java
Applet started.
```

Draw a rectangle in Java

```
 In this example, we draw a simple rectangle.
import java.awt.*;
import java.applet.Applet;
public class GrpRectEx extends Applet
{
 public void paint(Graphics grp)
 {
  grp.setColor(Color.blue);
  grp.drawRect(100, 50, 150, 150);
 }
}
/*
<applet code="GrpRectEx.class" width="400" height="400">
</applet>
*/
```

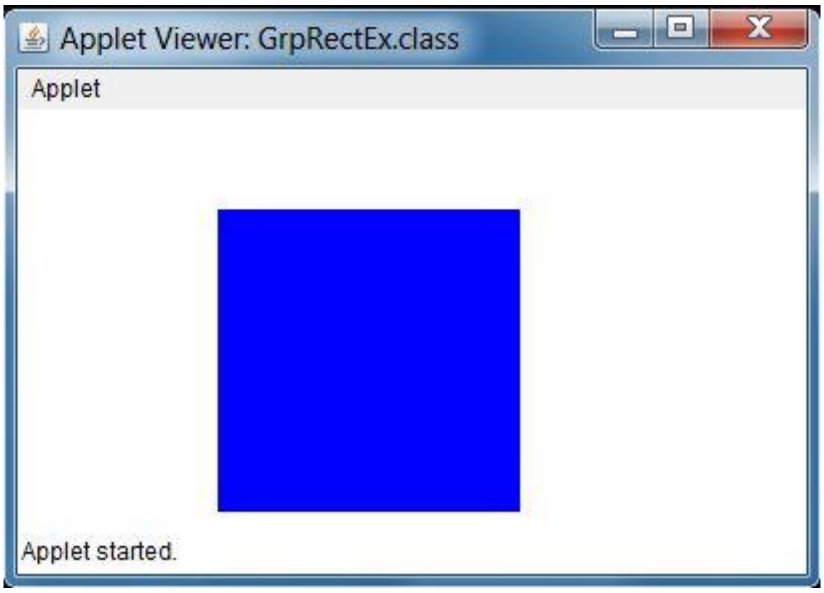
Output



Fill color in a rectangle in Java
 In this example, we fill in the color of our rectangle, usnig the grp.fillRect method.

```
import java.awt.*;
import java.applet.Applet;
public class GrpRectEx extends Applet
{
 public void paint(Graphics grp)
 {
  grp.setColor(Color.blue);
  grp.drawRect(100, 50, 150, 150);
  grp.fillRect(100, 50, 150, 150);
 }
}
/*
<applet code="GrpRectEx.class" width="400" height="400">
</applet>
*/
```

Output

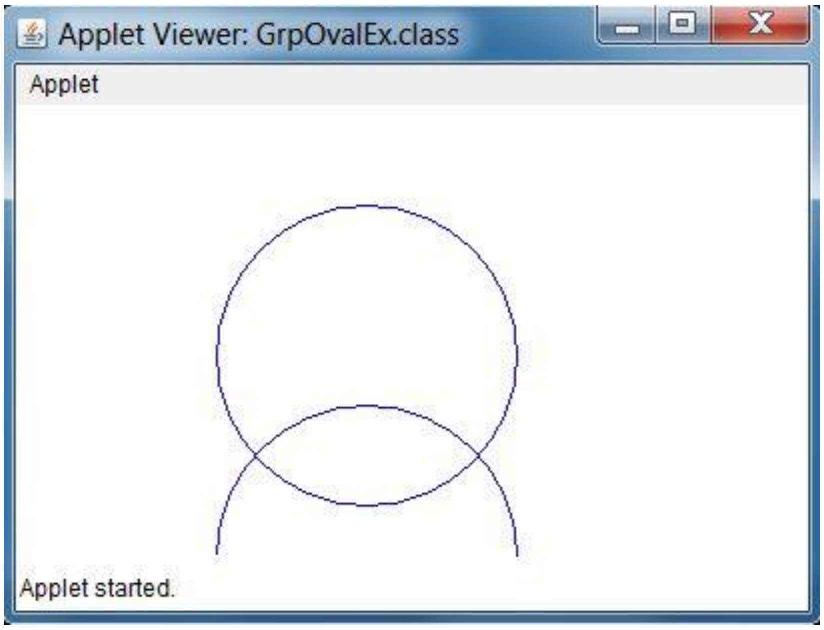


Draw an oval and an arc in Java

In this example, we create an oval and an arc with specified co-ordinates.

```
import java.awt.*;

import java.applet.Applet;

public class GrpOvalEx extends Applet

{

 public void paint(Graphics grp)

 {

  grp.setColor(Color.blue);

  grp.drawOval(100, 50, 150, 150);

  grp.drawArc(100, 150, 150, 150, 0, 180);

 }

}

/*

<applet code="GrpOvalEx.class" width="400" height="400">

</applet>

*/
```

Output



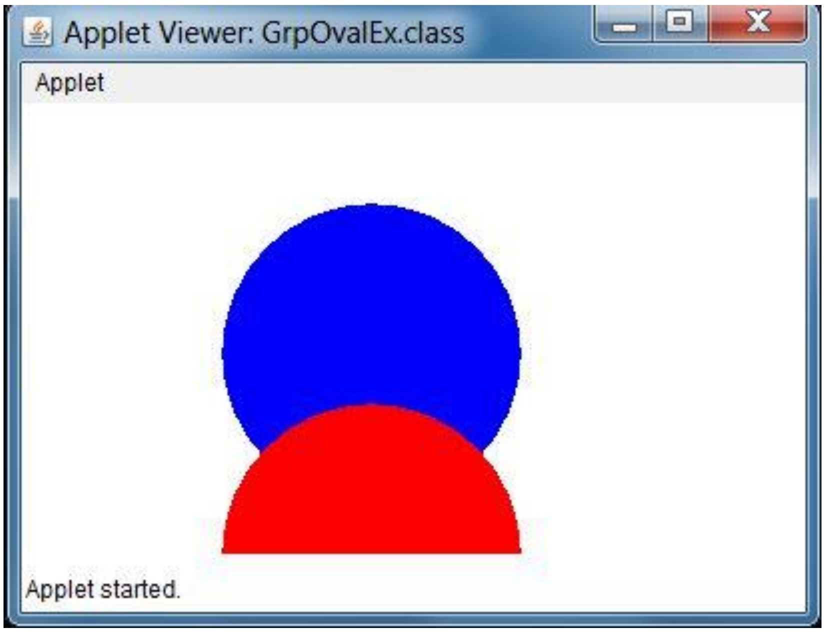
Fill color in oval and arc in Java

In this example, we fill an oval and an arc with a color, using the fillOval and fillArc methods.

```
import java.awt.*;

import java.applet.Applet;

public class GrpOvalEx extends Applet

{

 public void paint(Graphics grp)

  {

   grp.setColor(Color.blue);

   grp.fillOval(100, 50, 150, 150);

   grp.setColor(Color.red);

   grp.fillArc(100, 150, 150, 150, 0, 180);

  }

}

/*
```

```
<applet code="GrpOvalEx.class" width="400" height="400">

</applet>

*/
```

Output

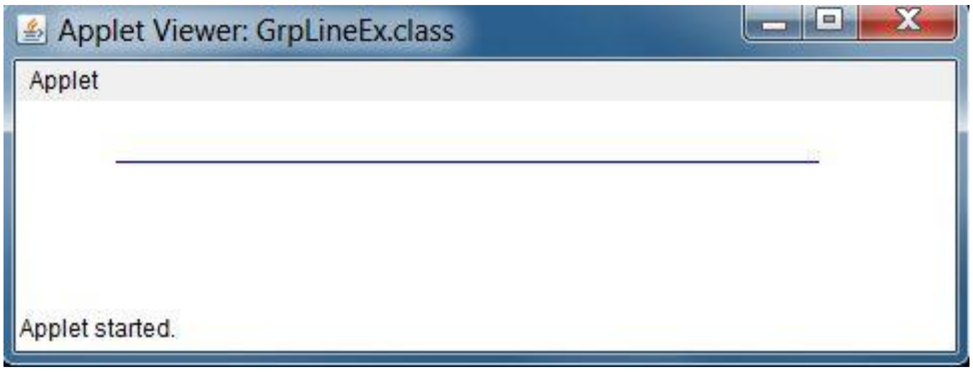


Draw a line in Java

In this example, we draw a simple line with specified co-ordinates.

```
import java.awt.*;
import java.applet.Applet;
public class GrpLineEx extends Applet
{
public void paint(Graphics grp)
{
grp.setColor(Color.blue);
grp.drawLine(50, 30, 400, 30);
}
}
/*
<applet code="GrpLineEx.class" width="400" height="400">
</applet>
*/
```

Output



Draw a Polygon in Java

A **polygon** is a closed figure with many lines joined one to another. The ending point of one line is the starting point to another line and finally the last point is joins with the first point. **java.awt.Graphics** class comes with two methods and one constructor to draw polygons.

1. **void drawPolygon(int x[], int y[], int numOfPoints):** Draws an **outline polygon** as per the coordinates specified in the **x[]** and **y[]** arrays. The **numOfPoints** gives the number of points (or to say, number of elements in the array) to join.
2. **void fillPolygon(int x[], int y[], int numOfPoints):** Draws a **solid polygon** as per the coordinates specified in the **x[]** and **y[]** arrays. The numOfPoints gives the number of points (or to say, number of elements in the array) to join.
3. **Polygon(int x [], int y [], int numOfPoints):** This constructor draws an **outline polygon** as per the coordinates specified in the **x[]** and **y[]** arrays. The **numOfPoints** gives the number of points to join.

All the above methods are used to draw polygons.

4 styles of drawing polygon are given.

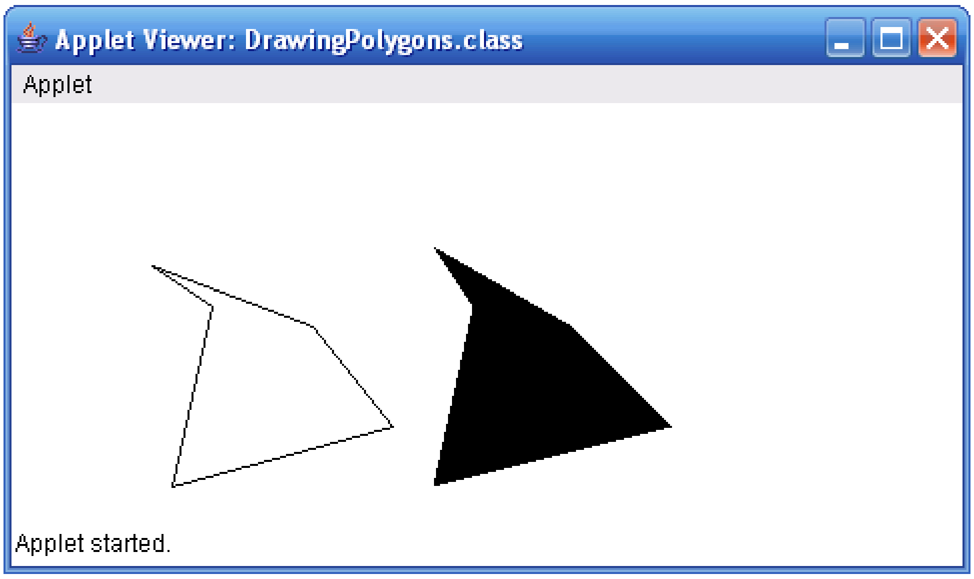
Using drawPolygon() with arrays.

Using java.awt.Polygon class.

Using drawLine() method.

Using addPoint() method.

```
import java.awt.*;
import java.applet.*;
public class DrawingPolygons extends Applet
{
  public void paint(Graphics g)
  {
    int x[] = { 70, 150, 190, 80, 100 };
    int y[] = { 80, 110, 160, 190, 100 };
    g.drawPolygon (x, y, 5);

    int x1[] = { 210, 280, 330, 210, 230 };
    int y1[] = { 70, 110, 160, 190, 100 };
    g.fillPolygon (x1, y1, 5);
  }
}
/*

<applet code="DrawingPolygons.class" width="350" height="300">
       </applet>
*/
```

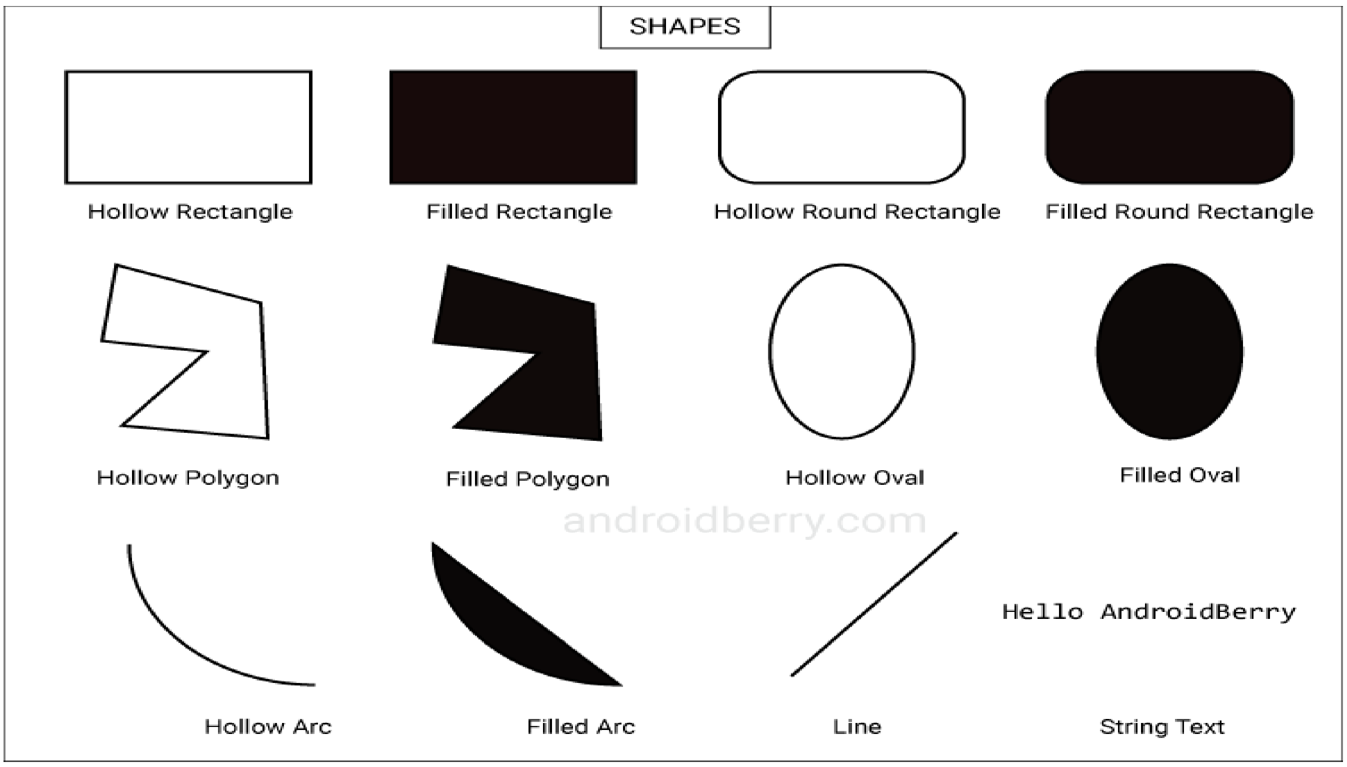


Graphics Class Summary

The graphics class of java includes methods for drawing different types of shapes,from line to polygons to text in a variety of fonts.

In applet, the paint() method and a Graphics class object is used to display text. To draw shapes on applet graphics class methods are used which of the have arguments representing ends points,corners in the coordinate system.

Drawing Shapes



Graphics Class method to draw shapes

Method Name	**Description**
drawRect()	This method draws a hollow rectangle
fillRect()	This method draws a filled rectangle
drawRoundRect()	This method draws a hollow rectangle with rounded corners.
fillRoundRect()	This method draws a filled rectangle with rounded corners.
drawPolygon()	This method draws a hollow polygon.
fillPolygon()	This method draws a filled polygon.

Method Name	Description
drawOval()	This method draws a hollow oval.
fillOval()	This method draws a filled oval.
drawArc()	This method draws a hollow arc.
fillArc()	This method draws a filled arc.
drawLine()	This method draws a straight line.
drawString()	This method displays a string text.

Font Class in Java Example

A font determines look of the text when it is painted. Font is used when painting text on a Graphics context and is a property of AWT Component.

All Fonts are stored into Font class.

Properties of Font class

- **Family Name -** It specifies the general name of the font.
- **Logical Name -** It specifies the category of the font.
- **Face Name -** It specifies specific font

How to set font?

1. To set the font to applet first of all create an object of Font class.

Syntax

```
Font fontObj = new Font (String FontName, int FontStyle, int
FontSize);
```

Explanation

- **FontName -** It specifies name of the font. Java Supports Dialog, DialogFont,Serif ,Sans Serif, Monospaced and Symbol.

- **FontStyle -** It specifies the different font style to font. It has 3 constants as

Constant	Description
Font.BOLD	It makes the font style bold.
Font.ITALIC	It makes the font style Italic.
Font.PLAIN	It make the font style plain.

Multiple font style can be applied by using OR | symbol Ex-Font.BOLD|Font.PLAIN

- **FontSize -** It specifies the font size in integers.

2. After creating an object java has setFont () method to set the font.

Syntax

```
void setFont (fontObj);
```

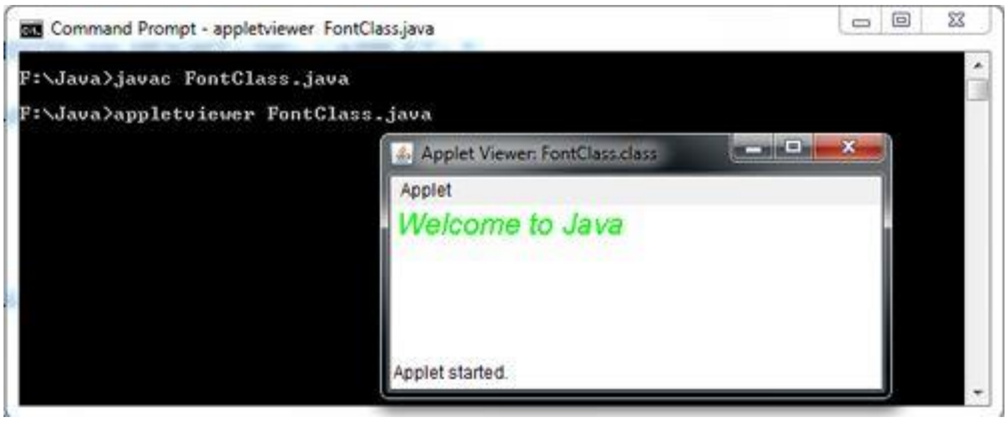
Example

```
Font myFont = new Font("Consolas",Font.BOLD,30);

setFont (myFont);
```

Example Font Class in Applet

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class FontClass extends java.applet.Applet
{
Font f;
String m;
public void init()
{
f=new Font("Arial",Font.ITALIC,20);
m="Welcome to Java";
setFont(f);
}
public void paint(Graphics g)
{
```

```
Color c=new Color(0,255,0);
g.setColor(c);
g.drawString(m,4,20);
}
}
/* <APPLET CODE ="FontClass.class" WIDTH=300 HEIGHT=200>
</APPLET> */
```



```
F:\Java>javac FontClass.java

F:\Java>appletviewer FontClass.java
```

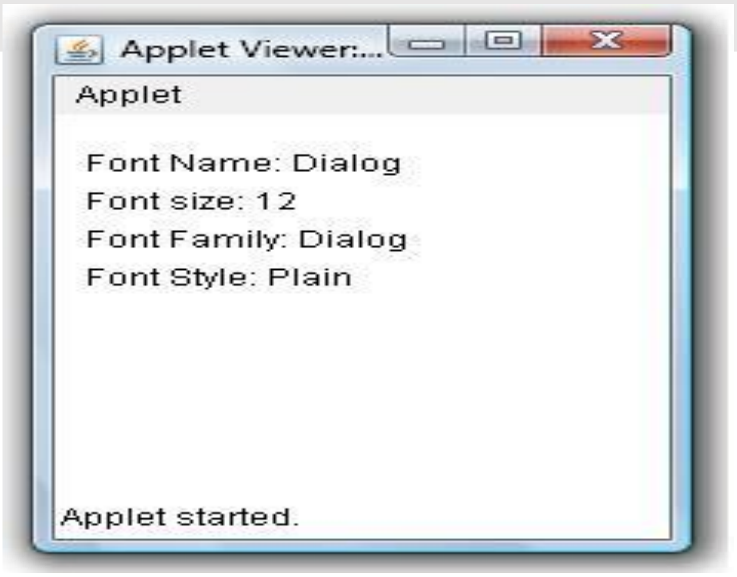
Command Prompt - appletviewer FontClass.java
Applet Viewer: FontClass.class
Applet
Welcome to Java
Applet started.

Get Font Information Example

```
/*
 Get Font Information Example
 This java example shows how to get Font information like font name,
 size, style using Java AWT Font class.
*/

import java.applet.Applet;
import java.awt.Font;
import java.awt.Graphics;



/*
<applet code="GetFontInformation" width=200 height=200>
</applet>
*/

public class GetFontInformation extends Applet{

 public void paint(Graphics g){

  /*
  * To get current font, use
  * Font getFont() method.
  */
```

```
Font currentFont = g.getFont();

/*
* To get a font name, use
* String getName()
* method of AWT Font class.
*/

String fontName = currentFont.getName();

/*
* To get font size, use
* int getSize()
* method of Font class.
*/

int size = currentFont.getSize();

/*
* To get style of font, use
* int getStyle()
* method of Font class.
*
* Style can be Font.BOLD, Font.ITALIC or Font.PLAIN.
*
* You can directly call isBold(), isItalic() or
* isPlain() methods of AWT Font class.
*/

int style = currentFont.getStyle();
String fontStyle = "";

if( (style & Font.BOLD) == Font.BOLD)
fontStyle = "Bold";

if( (style & Font.ITALIC) == Font.ITALIC)
fontStyle = "Italic";

if( (style & Font.PLAIN) == Font.PLAIN)
fontStyle = "Plain";

/*
* To get font family, use
* String getFamily()
* method of Font class.
*/

String family = currentFont.getFamily();
```

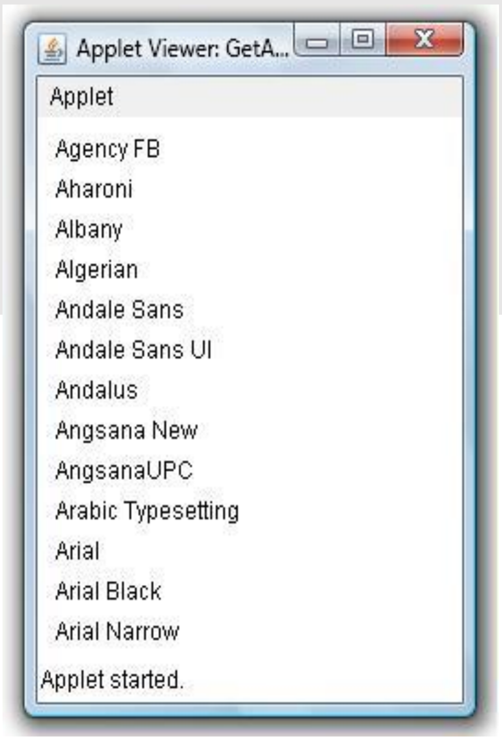
```
 g.drawString("Font Name: " + fontName, 10, 30);
 g.drawString("Font size: " + size, 10, 50);
 g.drawString("Font Family: " + family, 10, 70);
 g.drawString("Font Style: " + fontStyle, 10, 90);
 }
}
```



```
Font Name: Dialog
Font size: 12
Font Family: Dialog
Font Style: Plain
```

Get Available Font Family Names Example

```
/*
 Get Available Font Family Names Example
 This java example shows how to get available font family names using
 getAvailableFontFamilyNames method of GraphicsEnvironment class.
*/

import java.applet.Applet;
import java.awt.Graphics;
import java.awt.GraphicsEnvironment;


/*
<applet code="GetAvailableFonts" width=200 height=200>
</applet>
*/
public class GetAvailableFonts extends Applet{

 public void paint(Graphics g){

  GraphicsEnvironment graphicsEnvironment =
 GraphicsEnvironment.getLocalGraphicsEnvironment();


 String fontNames[] = graphicsEnvironment.getAvailableFontFamilyNames();
 int y = 20;
 for(int i=0; i < fontNames.length; i++){
 //print font names
 g.drawString(fontNames[i], 10, y);
 y =y+20;
 }
 }
}
```



```
Applet Viewer: GetA...
Applet
 Agency FB
 Aharoni
 Albany
 Algerian
 Andale Sans
 Andale Sans UI
 Andalus
 Angsana New
 AngsanaUPC
 Arabic Typesetting
 Arial
 Arial Black
 Arial Narrow
Applet started.
```

Click on following links to view complete Article

Applet Life Cycle Example

Basic Java Applet Example

Create Number counter in an Applet using Thread Example

Display Image in an Applet Example

Draw 3D Rectangle & Square in Applet Window Example

Draw Arc in Applet Window Example

Draw Dots at Random Locations in an Applet Example

Draw Line in Applet Window Example

Draw Oval & Circle in Applet Window Example

Draw Rectangle & Square in Applet Window Example

Draw Rounded Corner Rectangle & Square in Applet Window Example

Draw Smiley In Applet Example

Fill Rectangle & Square in Applet Window Example

Generate Bouncing Lines Using Applet Example

Get Applet Context Example

Get Applet Directory URL or Code Base Example

Get Applet Info Example

Get Applet parameter Example

Get Applet's Document URL or Directory Base Example

Get Background Color Of an Applet Window Example

Get Foreground Color Of an Applet Window Example

Hello World Applet Example

Load New HTML File Using Applet Context Example

Resize Applet Window Example

Set Background Color Of an Applet Window Example

Set Foreground Color Of an Applet Window Example

Set Status Message in Applet Window Example

Using Applet dimension to print center aligned text Example