

GANPAT UNIVERSITY  
U. V. PATEL COLLEGE OF ENGINEERING

# 2CEIT302

## OBJECT ORIENTED PROGRAMMING

UNIT 10

**APPLETS**

Prepared by: Prof. Y. J. Prajapati (Asst. Prof in IT Dept. , UVPCE)

# Outline



- Applet Basics, Methods of Building an Applet
- Displaying Text in Status Bar,
- The HTML Applet Tag, Reading Parameters into Applets, Colors in Applet, Getting Document base and Codebase
- Interfaces in Applet, Multimedia in Applet (Playing Audio Clips, Images in Applet, Applet Showing Other HTML Pages)

# Applet Basics

- ❑ An **applet** is a Java program that runs in a Web browser.
- ❑ It runs inside the browser and works at client side.
- ❑ Applet is embedded in a HTML page using the APPLET or OBJECT tag and hosted on a web server.
- ❑ Applets are used to make the web site more dynamic and entertaining.

## Important points :

- ❑ All applets are sub-classes (either directly or indirectly) of `java.applet.Applet` class.
- ❑ Applets are not stand-alone programs. Instead, they run within either a web browser or an applet viewer. JDK provides a standard applet viewer tool called applet viewer.
- ❑ In general, execution of an applet does not begin at `main()` method.
- ❑ Output of an applet window is not performed by `System.out.println()`. Rather it is handled with various AWT methods, such as `drawString()`.

# Java Applet vs Java Application

Java Application	Java Applet
Java Applications are the stand-alone programs which can be executed independently	Java Applets are small Java programs which are designed to exist within HTML web document
Java Applications must have main() method for them to execute	Java Applets do not need main() for execution
Java Applications just needs the JRE	Java Applets cannot run independently and require API's
Java Applications do not need to extend any class unless required	Java Applets must extend java.applet.Applet class
Java Applications can execute codes from the local system	Java Applets Applications cannot do so
Java Applications has access to all the resources available in your system	Java Applets has access only to the browser-specific services

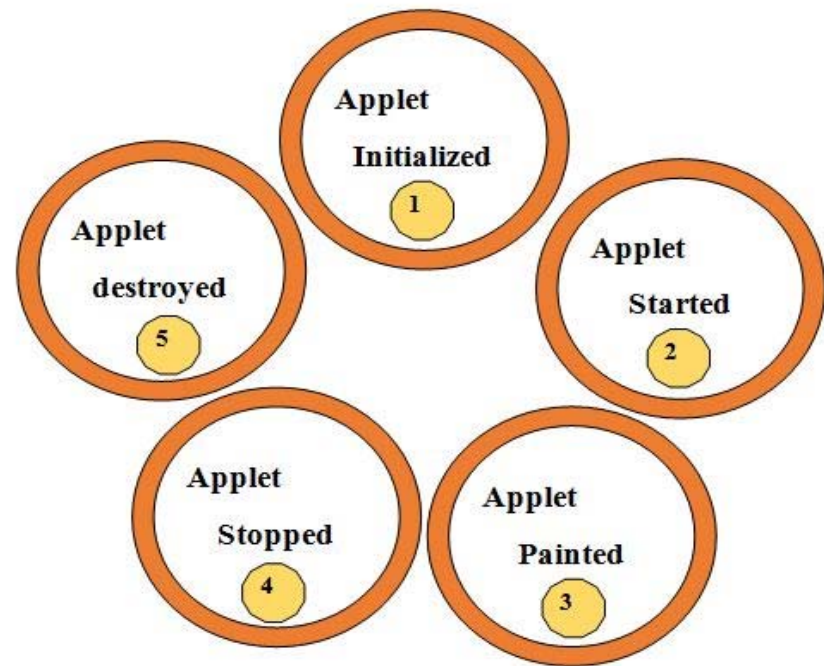
## Applet Hierarchy in Java

- class java.lang.**Object**
  - ▣ class java.awt.**Component**
    - class java.awt.**Container**
      - class java.awt.**Panel**
        - class java.applet.**Applet**

Java Applet class which is a class of applet package extends the Panel class of awt package. The Panel class is a subclass of the Container class of the same package. The Container class is an extension of Component class belonging to the same package. The Component class is an abstract class and derives several useful classes for the components such as Checkbox, List, buttons, etc.

# Life Cycle of an Applet

- ❑ Applet is initialized.
- ❑ Applet is started.
- ❑ Applet is painted.
- ❑ Applet is stopped.
- ❑ Applet is destroyed.



# Lifecycle methods for Applet

- **public void init():** is used to initialize the Applet. It is invoked only once.
- **public void start():** is invoked after the init() method or browser is maximized. It is used to start the Applet.
- **public void paint(Graphics g):** is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.
- **public void stop():** is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
- **public void destroy():** is used to destroy the Applet. It is invoked only once.

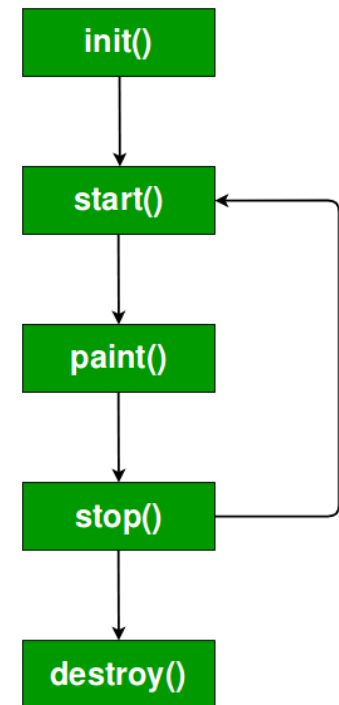
# Lifecycle methods for Applet

It is important to understand the order in which the various methods shown in the above image are called. When an applet begins, the following methods are called, in this sequence:

- `init( )`
- `start( )`
- `paint( )`

When an applet is terminated, the following sequence of method calls takes place:

- `stop( )`
- `destroy( )`





# How to run an Applet ?

There are two ways to run an applet

- By html file.
- By appletViewer tool (for testing purpose).

Simple example of Applet by html file:

- create an applet and compile it.
- After that create an html file and place the applet code in html file.

**//First.java**

```
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet
{
    public void paint(Graphics g){
        g.drawString("welcome",150,150);
    }
}
```

**myapplet.html**

```
<html>
<body>
<applet code="First.class" width="300" height="300">
</applet>
</body>
</html>
```

## Applet by appletviewer tool

To execute the applet by appletviewer tool, create an applet that contains applet tag in comment and compile it. After that run it by: appletviewer First.java.

```
//First.java
```

```
import java.applet.Applet;
```

```
import java.awt.Graphics;
```

```
public class First extends Applet
```

```
{
```

```
  public void paint(Graphics g)
```

```
  {
```

```
    g.drawString("welcome to applet",150,150);
```

```
  }
```

```
}
```

# Displaying Text in Status Bar

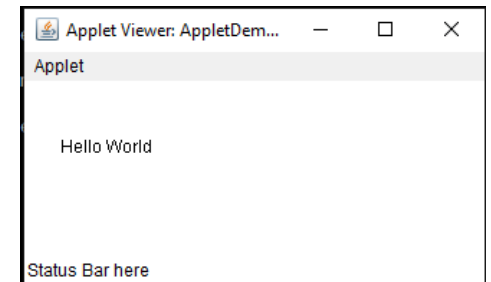
## AppletDemo.java

```
import java.applet.*;
import java.awt.*;

public class AppletDemo extends Applet {
    public void paint (Graphics g) {
        g.drawString ("Hello World", 25, 50);
        showStatus("Status Bar here");
    }
}
```


## Hello.html

```
<html>
  <title>The Hello, World Applet</title>
  <hr>
  <applet code = "AppletDemo.class" width = "320"
    height = "120">
    If your browser was Java-enabled, a "Hello, World"
    message would appear here.
  </applet>
  <hr>
</html>
```



# Displaying Graphics in Applet

- **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
- **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
- **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
- **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
- **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.
- **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).

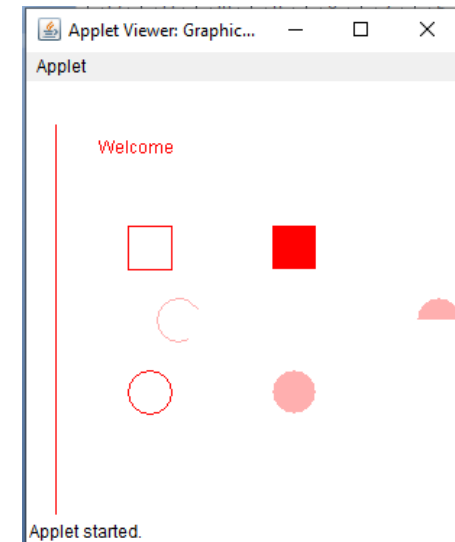
- 
- **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
  - **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
  - **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc.
  - **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
  - **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

# Example of Graphics in applet

```
import java.applet.Applet;
import java.awt.*;
public class GraphicsDemo extends Applet{
    public void paint(Graphics g){
        g.setColor(Color.red);
        g.drawString("Welcome",50, 50);
        g.drawLine(20,30,20,300);
        g.drawRect(70,100,30,30);
        g.fillRect(170,100,30,30);
        g.drawOval(70,200,30,30);
        g.setColor(Color.pink);
        g.fillOval(170,200,30,30);
        g.drawArc(90,150,30,30,30,270);
        g.fillArc(270,150,30,30,0,180);
    }
}
```

## Html file

```
<html>
<body>
<applet code="GraphicsDemo.class" width="300" height="300">
</applet>
</body>
</html>
```

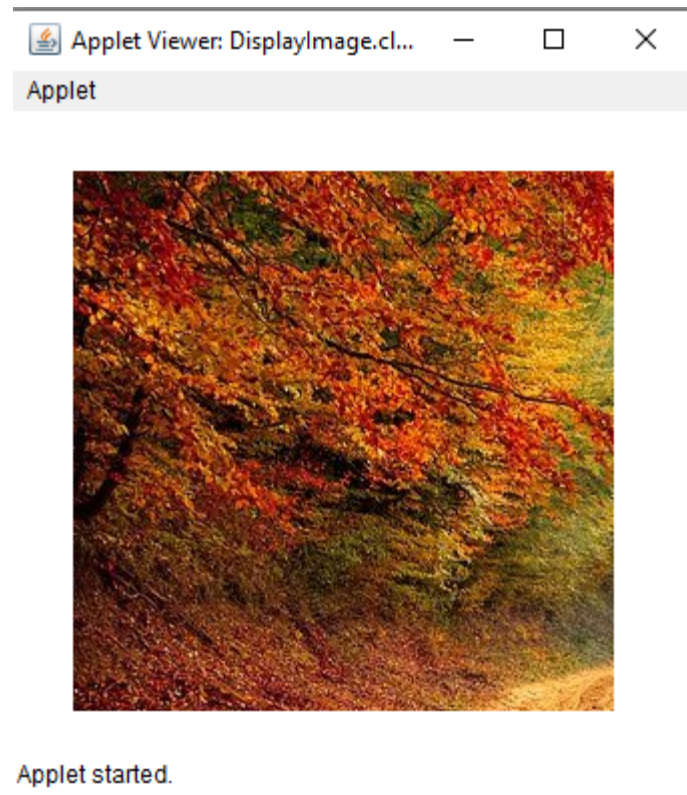


# Displaying Image in Applet

- Applet is mostly used in games and animation. For this purpose image is required to be displayed. The `java.awt.Graphics` class provide a method `drawImage()` to display the image.
- **public URL getDocumentBase():** is used to return the URL of the document in which applet is embedded.
- **public URL getCodeBase():** is used to return the base URL.

```
import java.awt.*;
import java.applet.*;
public class DisplayImage extends Applet {
    Image picture;
    public void init() {
        picture = getImage(getDocumentBase(),"sonoo.jpg");
    }
    public void paint(Graphics g) {
        g.drawImage(picture, 30,30, this);
    }
}
```

```
<html>
<body>
<applet code="DisplayImage.class" width="300" height="300">
</applet>
</body>
</html>
```





# Animation in applet

```
import java.awt.*;
import java.applet.*;
public class AnimationExample extends Applet {
    Image picture;
    public void init() {
        picture = getImage(getDocumentBase(),"butterfly.gif");
    }
    public void paint(Graphics g) {
        for(int i=0;i<50;i++){
            g.drawImage(picture, i,30, this);
            try{Thread.sleep(100);}catch(Exception e){}
        }
    }
}
```

```
<html>
<body>
<applet code="DisplayImage.class" width="300" height="300">
</applet>
</body>
</html>
```

