Applet basics

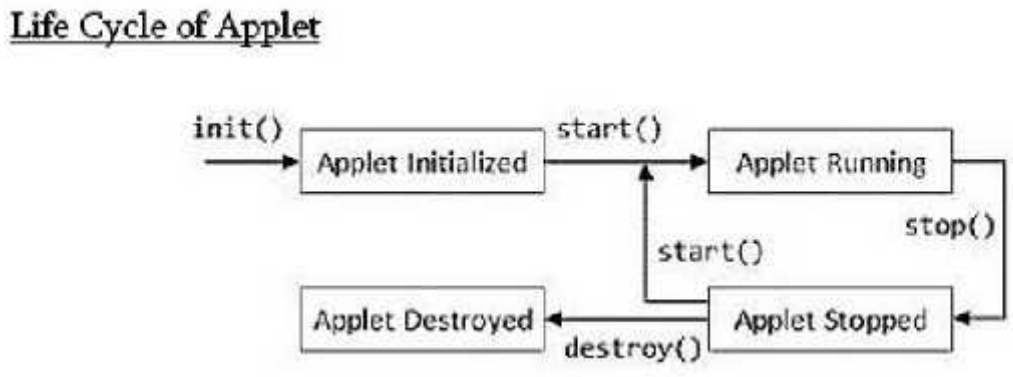
"An applet is a Java program that runs in a Web browser."

- All **applets** are subclass of ***java.applet.Applet*** class.
- Execution of an applet does not begin at main( ). Thus Applets do not need main( ) method.
- Output is handled with various AWT methods, such as drawString( ), which outputs a stringto a specified X,Y location
- *To use an applet, it is specified in an HTMLfile using applet tag as follows*

```
<applet code ="SimpleApplet" width=200 height=60>
 … … … … … …
</applet>
```

- Applets must be run under an applet viewer or a Java-compatible browser

Life cycle of an applet



Applet initialization and termination

- When an applet begins, the following methods are called, in this sequence:
  - init( )
  - start( )
  - paint( )
- an applet begins, the following methods are called, in this sequence:
  - stop( )
  - destroy( )

init( ) :
- The init( ) method is the first method to be called.
- Initializing the variables.
- This method is called only once during the run time of the applet.

start ( )
- Called after init( ).
- Also called to restart an applet after it has been stopped.
- Called each time an applet's HTML document is displayed onscreen.

paint( ):
- Called each time an applet's output must be redrawn.
- Also called when the applet begins execution.
- The method has one parameter of type Graphics.
- This parameter will contain the graphics context, which describes the graphics environment in which the applet is running.

stop( ):
- Called when a web browser leaves the HTML document containing the applet.
- Used to suspend threads that don't need to run when the applet is not visible.
- We can restart them when start( ) is called.

destroy( ):
- called when the environment determines that the applet needs to be removed completely from memory.
- The stop( ) method is always called before destroy( ).

Simple applet program

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

public class MyApplet extends Applet {
        public void paint(Graphics g){
                g.drawString("A Simple Applet", 20,20);
        }
}
```

- first import statement imports the Abstract Window Toolkit (AWT) classes
- The AWT contains support for a window-based, graphical user interface.
- ***The applet class must be declared as public***,
  - because it will be accessed by code that is outside the program.
- **The paint( ) method must be overridden by the applet:**
  - defined by the AWT

- paint( ) is called each time the applet must redisplay its output.
- **Situations When paint is called.**
  - When the applet begins execution.
  - The window in which the applet is running can be overwritten by another window and then uncovered.
  - The applet window can be minimized and then restored.
  - Whenever the applet must redraw its output, paint( ) is called.
- **Parameter**: Graphics object which describes the graphics environment.
- **drawString( )**:
  - is a member of the Graphics class.
  - This method outputs a string beginning at the specified location (X, Y).
- **general form:**

```
void drawstring(String message, int x, int y)
```

  - Here, message is the string to be output beginning at x, y.
  - In a Java window, the upper-left corner is location 0, 0.

Compilation and running

- In fact, there are two ways in which you can run an applet:
  - Executing the applet within a Java-compatible web browser.
  - Using an appletviewer tool.

1. running applet using applet-viever

- First create html code in the applet program and comment it with c type comment.
- Consider the following applet program say MyApplet.java

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

/*
<applet code="MyApplet" width=200 height=60>
<applet>*/

class MyApplet extends Applet {
 . . .
}
```

- Now you can run this program using appletviewer as shown below

```
C:\> javac MyApplet.java

C:\> appletviewer MyApplet.java
```

2. running applet in web browser

- To execute an applet in a web browser, you need to write a short HTML text file that contains a tag APPLET that loads the applet.
- Say file name as say ***RunApp.html***

```
<applet code = "MyApplet" width=200 height=60>
.....
</applet>
```

- The width and height specify the dimensions of the display area used by the applet.

```
import java.awt.*;
import java.applet.*;
import javax.swing.JOptionPane;

/*    <applet code = DemoApplet HEIGHT =200 WIDTH=200>
</applet>
*/

public class DemoApplet extends Applet {

 // First method to begin with.
public void init() {
      JOptionPane.showMessageDia log (null, " I nit Method");
  }

 //Second method. Called after init.
  public void start() {
    JOptionPane.showMessageDialog (null, "Start Method");
 }

  // Called when the applet is stopped.
   public void stop() {
    JOptionPane.showMessageDialog (null, "Stop Method");
        }

  // Called when applet is terminated.  last  method executed.
   public void destroy() {
    JOptionPane.showMessageDialog (null, "Destroy Method");   }

  // Called when an applet's window must be restored.
   public void paint(Graphics g) {
      g.drawString ( "Paint is ca lled", 20, 20);
   }
}
```

Passing parameters to applets

- APPLET tag in HTML allows you to pass parameters to your applet.
- To retrieve a parameter, use the getParameter( ) method
- General form
  - String ***getParameter***(String paramName)
- It returns the value of the specified parameter in the form of a String object.
- Thus, for numeric and boolean values, you will need to convert their string representations into their internal formats.
- Here is an example that demonstrates passing parameters:

//Code of PARAM_Test.html

```
<html>
<title>PARAM_test</title>
<applet code="PARAM_test.class" width=400 height=400>
<PARAM NAME="name" VALUE="Achin Jain">
</applet>
</html>
```

//Code of PARAM_Test.java

```
import java.applet.*;
import java.awt.*;
public class PARAM_test extends Applet
{
    String str=null;
    public void init()
    {
        str=getParameter("name");
    }
    public void paint(Graphics g)
    {
        g.drawString(str,100,100);
    }
}
```

PARAM Name is used as "*name*" which should be passed as parameter in getParameter() method to fetch the value of the parameter

Requesting Repainting

- Awt- based applet writes to its window only when its *paint()* method is called by the run-time system.
- Whenever an applet needs to update the information displayed in its window, it simply calls *repaint()*.
  - *void repaint()*

Using status window

- An applet can also output a message to the status window of the browser or applet viewer on which it is running
- **General form**

- **void showStatus(String msg)**

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

/*<applet code="StatusWin" width=300 height=50>
</applet>*/

public class StatusWin extends Applet{
        public void paint(Graphics g){
        g.drawString("This is applet window",10,20);
        showStatus("This is shown in the status window")
        }
}
```