Applet

Introduction

- An applet is a Java program that runs in a Web browser.
- An applet can be a fully functional Java application because it has the entire Java API at its disposal.

Applet vs Standalone java app

- An applet is a Java class that extends the java.applet.Applet class.
- A main method is not invoked on an applet, and an applet class will not define main.
- Applets are designed to be embedded within an HTML page.
- When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine.
- A JVM is required to view an applet. The JVM can be either a plug-in of the Web browser or a separate runtime environment.
- The JVM on the user's machine creates an instance of the applet class and invokes various methods during the applet's lifetime.
- Applets have strict security rules that are enforced by the Web browser.
- Other classes that the applet needs can be downloaded in a single Java Archive *JAR file.*

Life cycle of Applet

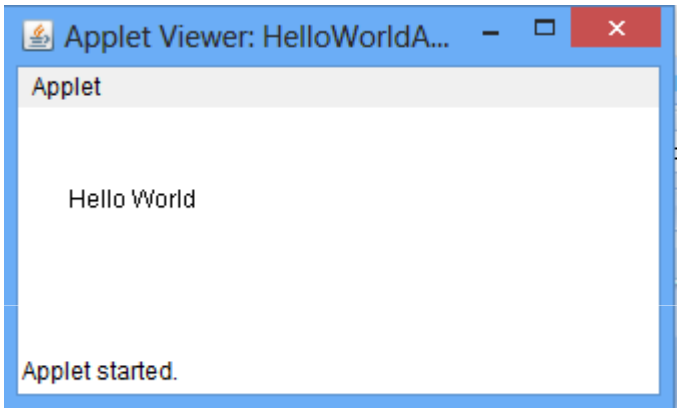
- **init:** This method is intended for whatever initialization is needed for your applet. It is called after the param tags inside the applet tag have been processed.
- **start:** This method is automatically called after the browser calls the init method. It is also called whenever the user returns 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.

Example

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

public class HelloWorldApplet extends Applet
{
  public void paint (Graphics g)
  {
    g.drawString ("Hello World", 25, 50);
  }


}
/*
<applet code="HelloWorldApplet.class" width="320" height="120">
</applet>
*/
```



Compilation: javac HelloWorldApplet.java
Execution: appletviewer HelloWorldApplet.java

Applet class

- The base Applet class provides methods that a derived Applet class may call to obtain information and services from the browser context.
- These include methods that do the following:
  - Get applet parameters
  - Get the network location of the HTML file that contains the applet
  - Get the network location of the applet class directory
  - Print a status message in the browser
  - Fetch an image
  - Fetch an audio clip
  - Play an audio clip
  - Resize the applet

Applet class (cont..)

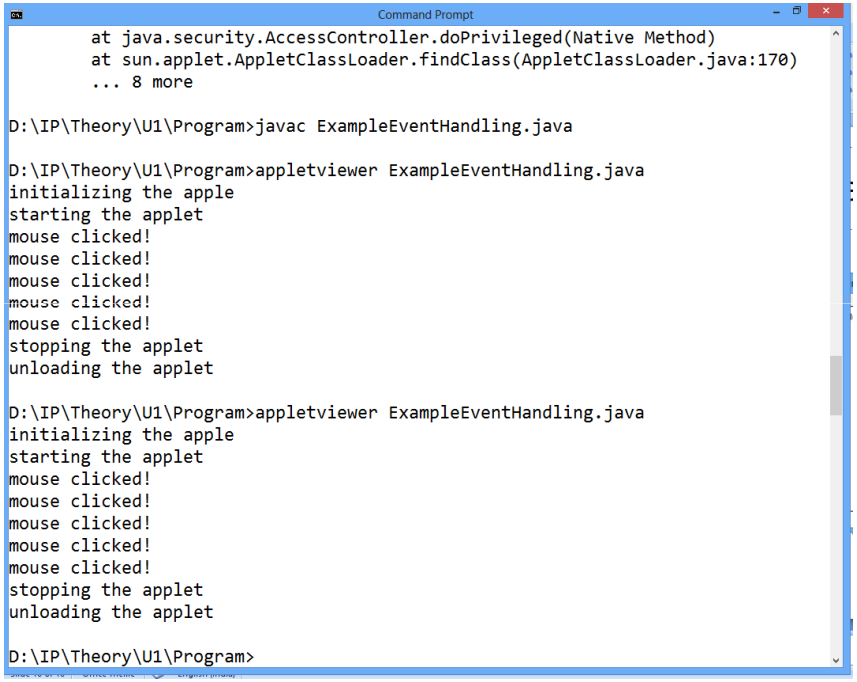
- Additionally, the Applet class provides an interface by which the viewer or browser obtains information about the applet and controls the applet's execution.
- The viewer may:
  - request information about the author, version and copyright of the applet
  - request a description of the parameters the applet recognizes
  - initialize the applet
  - destroy the applet
  - start the applet's execution
  - stop the applet's execution

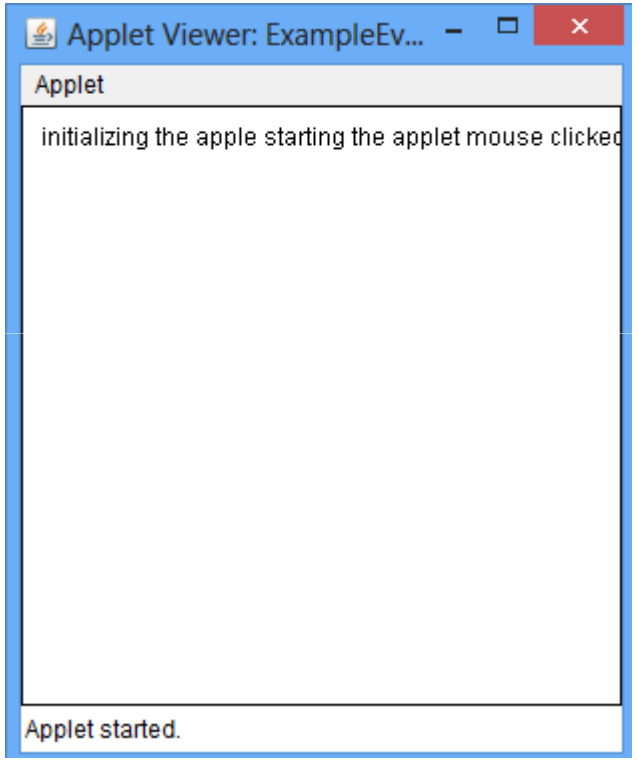
Event handling

```
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.applet.Applet;
import java.awt.Graphics;
public class ExampleEventHandling extends Applet
implements MouseListener
{
StringBuffer strBuffer;
public void init() {
addMouseListener(this);
strBuffer = new StringBuffer();
addItem ("initializing the apple ");
}
public void start() {
addItem ("starting the applet ");
}
public void stop() {
addItem ("stopping the applet ");
}
public void destroy() {
addItem ("unloading the applet");
}
```

```
void addItem (String word) {
System .out.println(word);
strBuffer.append(word);
repaint();
}
public void paint(Graphics g) {
//Draw a Rectangle around the applet's display
      area.
g.drawRect(0, 0,
getWidth() - 1,
getHeight() - 1);
//display the string inside the rectangle.
g.drawString(strBuffer.toString(), 10, 20);
}
public void mouseEntered(MouseEvent event)
      {
}
public void mouseExited(MouseEvent event) {
}
public void mousePressed(MouseEvent event)
      {
}
public void mouseReleased(MouseEvent
      event) {
}
public void mouseClicked(MouseEvent event) {
addItem ("mouse clicked! ");
}
}
/*
<applet code="ExampleEventHandling.class"
width="300" height="300">
</applet>
*/
```

output


Command Prompt
at java.security.AccessController.doPrivileged(Native Method)
at sun.applet.AppletClassLoader.findClass(AppletClassLoader.java:170)
... 8 more
D:\IP\Theory\U1\Program>javac ExampleEventHandling.java
D:\IP\Theory\U1\Program>appletviewer ExampleEventHandling.java
initializing the apple
starting the applet
mouse clicked!
mouse clicked!
mouse clicked!
mouse clicked!
mouse clicked!
stopping the applet
unloading the applet
D:\IP\Theory\U1\Program>appletviewer ExampleEventHandling.java
initializing the apple
starting the applet
mouse clicked!
mouse clicked!
mouse clicked!
mouse clicked!
mouse clicked!
stopping the applet
unloading the applet
D:\IP\Theory\U1\Program>


Applet Viewer: ExampleEv...
Applet
initializing the apple starting the applet mouse clicked
Applet started.