IT 222: Java Programming – II

Course Objectives:

- This course aims to develop students with knowledge with advanced features of java.
- Basic knowledge of programming in Java is expected from students.
- This course should be associated with laboratory experiments to augment the concepts taught in the class.

Course Description

- The Applet Class
- Even handling
- Introducing the AWT
- Using AWT controls
- Layout Managers, and Menus
- Introducing Swing,
- Enterprise Application Architectures,
- JDBC, XML and Java,
- Servlet Programming,
- JSP Programming

Unit 1: The Applet Class

- Applet basics
- An applet skeleton
- Two types of applets
- The HTML applet tag
- Passing parameters to applets

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.
- There are some important differences between an applet and a standalone Java application, including the following

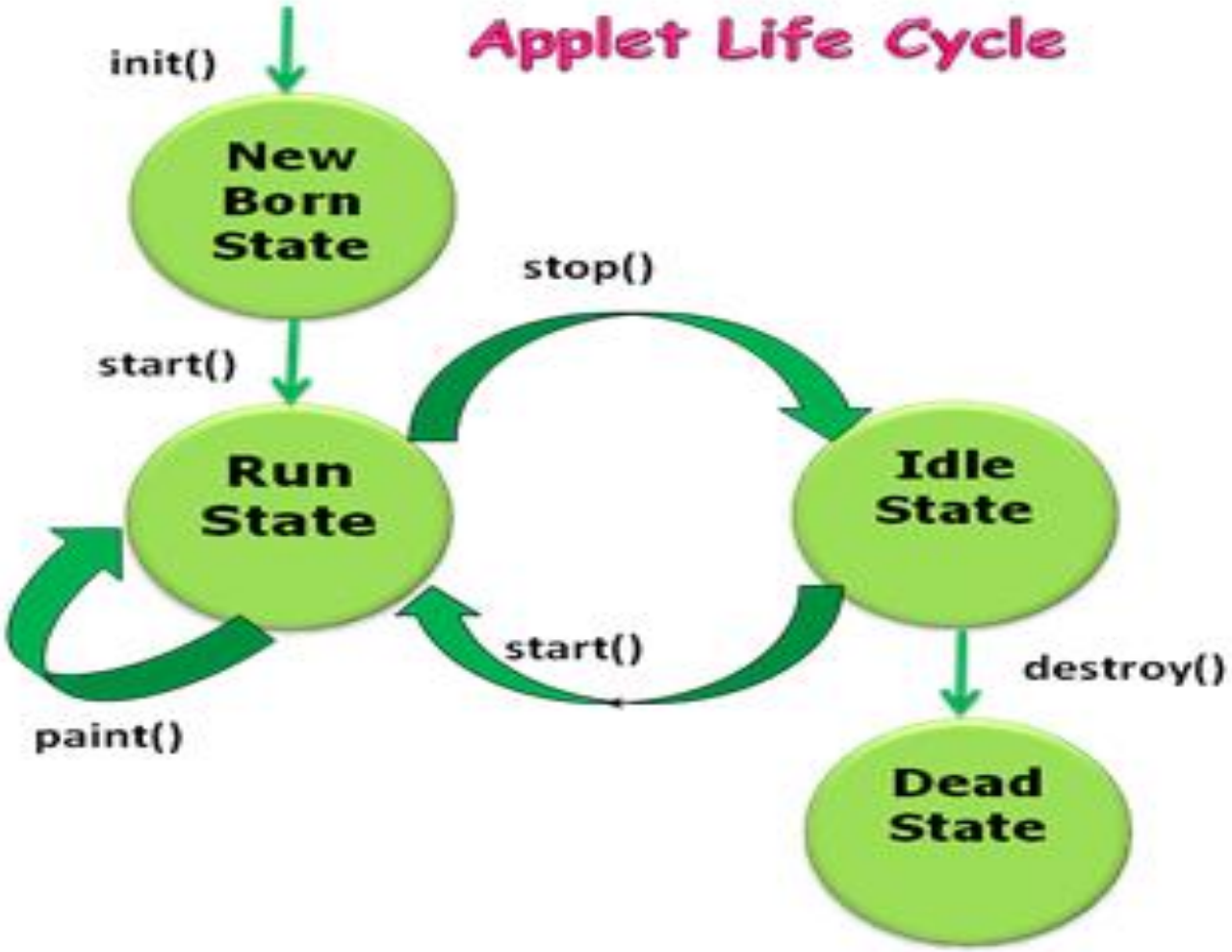
- An applet is a Java class that extends the java.applet.Applet class.
- A main() method is not invoked on an applet, and an applet class will not define main().
- Applets are designed to be embedded within an HTML page.
- When a user views an HTML page that contains an applet, the code for the applet 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. The security of an applet is often referred to as sandbox security, comparing the applet to a child playing in a sandbox with various rules that must be followed.
- Other classes that the applet needs can be downloaded in a single Java Archive (JAR) file.

summary

- An applet is a java program that runs in web browser
- An applet is a Java class that extends the java.applet.Applet class
- No main() method is used
- JVM is required to view an applet
- Applet have strict security rules that are enforced by web browser.

Life Cycle of an Applet



Life Cycle of an Applet

- Four methods in the Applet class gives you the framework on which you build any 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.

How to create First Applet

```
mport java.applet.*;
Import java.awt.*;
public class  image_load extends Applet
{
  public void paint(Graphics g)
  {
    g.drawString("Hello World", 100, 100);
  }
}
```

Html code

```
<html>
  <title>my first applet</title>
<applet  code="image_load.class" width="400"
   height="400">
</applet>
</html>
```

Types of applets

- A special type of Java program that runs in a Web browser is referred to as **Applet**. It has less response time because it works on the client-side.
- It is much secured executed by the browser under any of the platforms such as Windows, Linux and Mac OS etc.
- There are two types of applets that a web page can contain.
- **Local Applet**
- **Remote Applet**

Local applets

- **Local Applet** is written on our own, and then we will embed it into web pages.
- Local Applet is developed locally and stored in the local system.
- A web page doesn't need the get the information from the internet when it finds the local Applet in the system.
- It is specified or defined by the file name or pathname.
- There are two attributes used in defining an applet, i.e., the **codebase** that specifies the path name and **code** that defined the name of the file that contains Applet's code.

Specifying Local applet

Let's take an example of Local applet to understand how we can create it and embedded it into web page.

- First, we will create a Local Applet for embedding in a web page.
- After that, we will add that Local Applet to the web page.

<applet

codebase = "tictactoe"

code = "FaceApplet.class"

width = 120

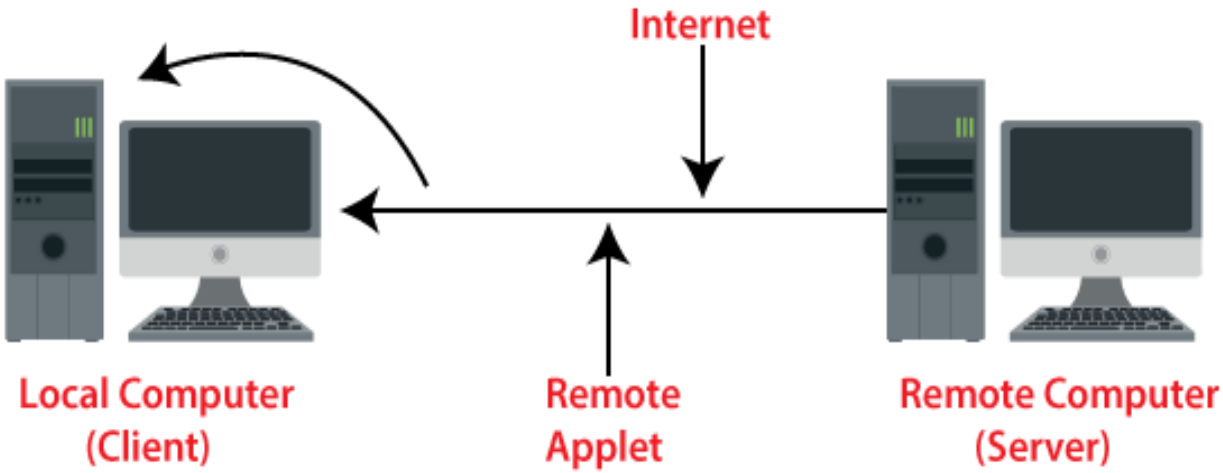
height = 120>

</applet>

```
//Import packages and classes
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.awt.event.*;
//Creating FaceApplet class that extends Applet
public class FaceApplet extends Applet
{
   //paint() method starts
   public void paint(Graphics g){
      //Creating graphical object
      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);
   }
}
```

Remote Applet

- A remote applet is designed and developed by another developer.
- It is located or available on a remote computer that is connected to the internet.
- In order to run the applet stored in the remote computer, our system is connected to the internet then we can download run it.
- In order to locate and load a remote applet, we must know the applet's address on the web that is referred to as Uniform Recourse Locator(URL).


Internet
Local Computer
(Client)
Remote
Applet
Remote Computer
(Server)

```
<applet
   codebase = "http://www.myconnect.com/app
   lets/"
   code = "FaceApplet.class"
   width = 120
   height =120>
</applet>
```

Differences

Local Applet	Remote Applet
There is no need to define the Applet's URL in Local Applet.	We need to define the Applet's URL in Remote Applet.
Local Applet is available on our computer.	Remote Applet is not available on our computer.
In order to use it or access it, we don't need Internet Connection.	In order to use it or access it on our computer, we need an Internet Connection.
It is written on our own and then embedded into the web pages.	It was written by another developer.
We don't need to download it.	It is available on a remote computer, so we need to download it to our system.

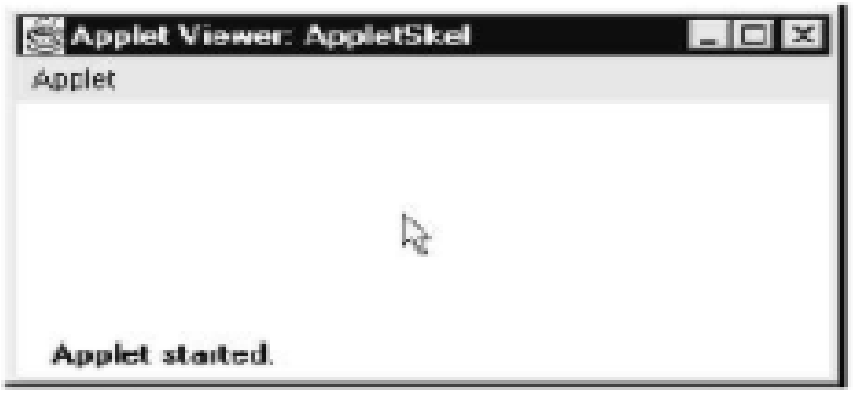
An Applet Skeleton

- All but the most trivial applets override a set of methods that provides the basic mechanism by which the browser or applet viewer interfaces to the applet and controls its execution.
- Four of these methods—init( ), start( ), stop( ), and destroy( )—are defined by Applet. Another, paint( ), is defined by the AWT Component class.
- Default implementations for all of these methods are provided.
- Applets do not need to override those methods they do not use.
- However, only very simple applets will not need to define all of them.
- These five methods can be assembled into the skeleton shown here:

```
// An Applet skeleton.
import java.awt.*;
import java.applet.*;
/*
<applet code="AppletSkel" width=300 height=100>
</applet>
*/
public class AppletSkel extends Applet {
// Called first.
public void init() {
// initialization
}
/* Called second, after init(). Also called whenever
the applet is restarted. */
public void start() {
// start or resume execution
}
// Called when the applet is stopped.
public void stop() {
// suspends execution
}
```

- /* Called when applet is terminated. This is the last method executed. */
  public void destroy() {
  // perform shutdown activities
  }
  // Called when an applet's window must be restored.
  public void paint(Graphics g) {
  // redisplay contents of window
  }
  }

- Although this skeleton does not do anything, it can be compiled and run.
- When run, it generates the following window when viewed with an applet viewer:



- **Applet Initialization and Termination**: It is important to understand the order in which the various methods shown in the skeleton are called.
- When an applet begins, the AWT calls the following methods, in this sequence:
  1. init( )
  2. start( )
  3. paint( )
- When an applet is terminated, the following sequence of method calls takes place:
  1. stop( )
  2. destroy( )

Applet HTML Tag

Syntax:

```
<applet code="URL" height="200" width="100"
   >.....</applet>
```

Attributes

Specific Attributes

Attribute name	Value	Description
code	URL	It specifies the URL of Java applet class file.
width	pixels	It specifies the display width of the applet panel.
height	pixels	It specifies the display height of applet panel
align	o left o right o top o middle o bottom	It specifies the position of applet application relative to surrounding content.

alt	text	It is used to display alternative text in case browser does not support Java.
archive	URL	This specifies the archived or compressed version of an applet application.
object	name	It specifies the URL or reference to a serialized representation of an applet.
codebase	URL	It specifies the exact or relative URL of applets .class file specified in the code attribute.
hspace	pixels	It specifies the horizontal space around the applet.
vspace	pixels	It specifies the vertical space around the applet.
name	name	It specifies the name for the applet

Passing parameters to applets

- 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

```
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);
    }

}
```

Html file

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