Applet

The Java program can be broadly classified into two categories-

i) Application program

ii) Applet program

To create a user interactive application program, we use the Command-line arguments, which are restricted and not very user friendly. A better alternative is to use forms to interact with the user. To accept data from the user, we can create applets in Java. Java Provides Java Development Kit (JDK) that enables to create user interactive forms in applets.

JDK Consists of a package called Abstract Window Toolkit (AWT). AWT is an application Programming Interface (API) that is responsible for building the Graphical User Interface (GUI) in Java and its Consists of a collection of classes and methods that enables us to design and manage the Graphical User Interface (GUI) applications. The AWT package supports applets, which help in creating containers, such as frams or panels that run in the GUI environment.

Applet :

An applet is a Java program that can be embedded in an HTML Web page. It can be transported over the Internet from one computer to another and run using the Java enabled web browser or an applet viewer. Applet can perform arithmetic operation, display graphics, play sounds, accept user input, create animation and etc. Applets are ~~exceptionally~~ developed to supports the GUI in Java.

Advantages of Java Applet :

i) Java applets are written entirely in Java, so it automatically inherits the platform independence nature of the language.

ii) Most of the web browsers are applet Compatible.

iii) Java library has an extensive set of classes and methods, which facilitate to write any GUI program using applets.

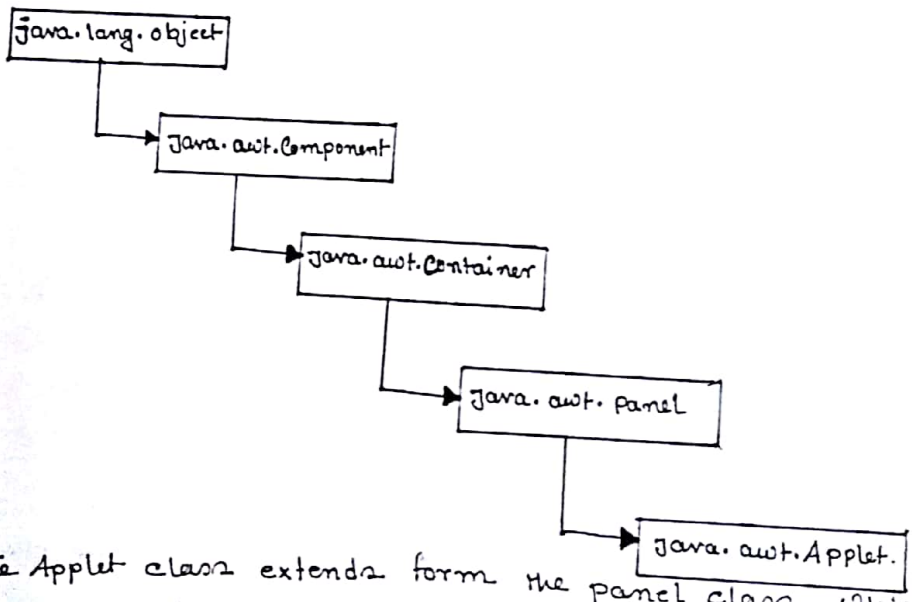
Local and Remote Applet :-

Local Applet :- An applet developed locally and stored in a local system is known as a local applet. So, it does not require the Internet connection.

Remote Applet :- An applet is developed by someone and stored on a remote computer connected to the Internet. If our system is connected to the internet, we can download the remote applet onto our system via at the Internet and run it.

The Applet class :-

The Applet class is a member of the Java API package, java.applet. We use the Applet class to create a Java program that displays an applet. The hierarchy of applet class is given below.



The Applet class extends form the panel class, which further extends from Container, Component and object classes. The object class is the member of java.lang package and is located at the top hierarchy of all the Java packages.

The component, container and panel classes are the members of the java.awt package and provides components, such as label, button or text fields. The Applet class is the only member of the java.applet package.

The Applet class contains various methods that are used to display text and image, play an audio file and responds when we interact with an applet. The following table lists the various method of the Applet class.

Method	Function.
void init()	Begins the execution of an applet when the web browser or Java tool calls it.
void start()	starts the execution of an applet.
String getAppletInfo()	Returns a string that describes the information about an applet.
boolean isActive()	Returns the boolean type value, true if the applet is started otherwise returns false.
void stop()	Suspends the execution of an applet.
void resize(int width, int height)	changes the size of an applet, according to the specified height and width.
void resize(Dimension d)	changes the size of an applet according to the specified dimensions by 'd'.
Image getImage(URL, url)	Returns an image object that encapsulates the image found at the location specified by the URL.
void destroy()	Remove the applet completely from the memory.
void paint()	Redraws the applet's output.
void update()	calls the paint() method to redraw the drawing area.
void repaint()	Redraws an applet by calling the update() method.

* Creating an Applet :-

An applet is a program that executes in a Java-enabled browser or in a Java Development tool Kit (JDK), such as appletviewer. The JDK consists of a library of standard classes and utilities to build, test and document the Java programs. Some JDK utilities are :-

JDK Utilities.	Description.
Javac	converts Java source code to bytecode.
Java	Executes Java applications directly from the class file.
Appletviewer	Executes Java applets hosted by an HTML file.
Javadoc	creates an HTML documents based on Java source code.

To create an applet, we need to follow these steps:-

i) create a Java program for the Applet. (.java file)

ii) Compile the Java program (.class file).

iii) create a web page that contains an applet. (using <APPLET> tag)

iv) Run the applet.

eg:- To create an applet that displays text.

```
import java.awt.*;                  // importing the awt package.
import java.awt.applet.*;           // importing the applet package.

Public class japplet extends Applet
 {
      Public void paint (Graphics g)
          {
             g.drawString ("Hello, This is my first Applet", 50,100);
          }
 }
```

In the above code, the japplet class imports the java.applet and the java.awt packages to include the declaration of various built-in methods that are required for running an applet. The extends Keyword inherits the properties of the Applet class. We need the Graphics class to write in an applet. The japplet class contains the paint() method that overrides the default behavior of the Graphics class. We use the drawString() method of the Graphics class within the paint() method to display text on an

The drawString() method consists of three arguments. The first argument represents the String that we want to display on an applet. The second and third arguments represent the starting co-ordinates of x and y-axis for displaying the text at the desired position on an applet.

We need to save the above Java program as japplet.java and compile the program to create its class file. You create an HTML file to run the applet that embeds an applet in the web page by using the <APPLET> tag. We need to save the HTML file as japplet.html.

The HTML code for the above program is

```
<HTML>
<HEAD> <TITLE> My first Applet program </TITLE> </HEAD>
<BODY>
<APPLET CODE = "japplet.class" HEIGHT = 300 WIDTH = 250>
</APPLET>
</BODY>
</HTML>
```

In the above code, the <APPLET> tag contains the 'CODE' attribute, which embeds the class file of the HTML file. The attributes HEIGHT and WIDTH, set the dimension of the applet window.

Note:- If we want to view the output in an appletviewer, then the following command should be used —

appletviewer filename.html.

eg: appletviewer japplet.html.

Compile and Execute the Application.

The command to compile the japplet class is:

javac japplet.java.

The command to execute the application is:

appletviewer japplet.java.

CODE and CODEBASE :

The CODE attribute is used to indicate the name of the class file that holds the current Java applet. The CODE attribute is used when both the .java file and the .html file are located in the same directory.

The CODEBASE Attribute indicates the pathname where the .class file is stored. The CODEBASE attribute is used, if you store a java file in a directory different from an HTML file. The CODE attribute specifies the name of the class file whereas the CODEBASE attribute contains an alternate pathname where the classes are stored.

Passing Parameters to Applets :

We can pass parameters to an applet by using <PARAM> tag. The PARAM tag contains the NAME and VALUE attributes. For example we can define a parameter to indicate the color of the text, font and size of the text by using the <PARAM> tag.

eg:
```
<APPLET CODE = "japplet.class" HEIGHT = 25 WIDTH = 25>
<PARAM NAME = "color" VALUE = "Pink">
<PARAM NAME = "font" VALUE = "CourierNew">
<PARAM NAME = "size" VALUE = "10">
</APPLET>
```

Applet Life Cycle :

The life cycle of an applet describes the sequence of stages, which begin when an applet is loaded in an appletviewer or a web browser and ends when the applet is destroyed.

An applet inherits the properties and methods of the Applet class. Java provides init(), start(), stop(), paint() and destroy() as the basic applet methods to control the execution of an applet. The different stages of an applet life cycle are:-

- Initializing an applet
- starting the applet.
- stopping the applet.
- Destroying the applet.

i) Initializing an Applet :

The init() method initializes an applet when the applet is loaded for first time. It defines the objects and variables that are required to execute an applet. We apply the settings for fonts, colors, and initial parameters, such as variables and constants in the init() method. The init() method is also used to add components, such as buttons and check boxes to an applet.

Syntax of define the init() method :

```
Public void init()
{
    /* method definitions */
}
```

Note:- The initialization occurs only once in the applet's life cycle.

ii) Starting an Applet :

The start() method is called to start the execution of an applet after it's initialized with the init() method. The start() method can be called more than once in an applet.

Syntax of define the start() method :

```
public void start()
{
    /* method definitions */
}
```

iii) Stopping an Applet :

The stop() method suspends the execution of an applet. The stop() method is called when either an end user stops an applet or an applet loses the focus. We can use the stop() method to reset the variables and stop a running applet.

Syntax of defining the stop() method:

```
Public void stop()
{
    /* method definitions */
}
```

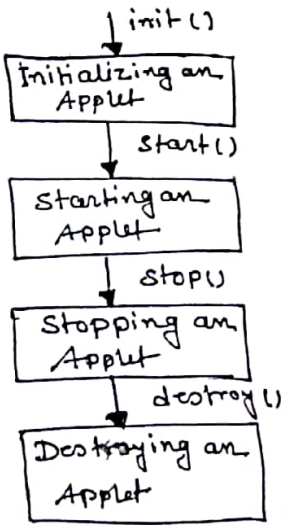
iv) Destroying an Applet :

The destroy() method is called when an applet is destroyed. When we want to exit from the web browser or appletviewer of Java, an applet calls this method to release the resources, such as parameters and images. This method occurs only once in the life cycle of an applet.

Syntax of define the destroy() method :

```
Public void destroy()
{
    /*method definitions */
}
```

The following figure shows the life cycle of an applet :



Life cycle of an Applet.

Or