Que. What is applet ? Explain with example.

Ans. An applet is an intelligent program designed to be transmitted over the Internet and executed by a Java-compatible Web browser. An applet is actually a tiny java program, dynamically downloaded across the network, just like an image, sound file or video clip.

An applet is not just an animation or media file. In other words, an applet is a program that can react to user input and dynamically change-not just run the same animation or sound over and over.

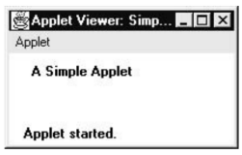
There are two ways in which you can run an applet:

➔Executing the applet within a Java-compatible Web browser.

➔Using an applet viewer tool. An applet viewer executes applet in a window.

```
import java.awt.*;
import java.applet.*;
/*
<applet code="SimpleApplet" width=200 height=60>
</applet>
 */
public class SimpleApplet extends Applet
{
    public void paint(Graphics g)
    {
     g.drawString("A Simple Applet", 20, 20);
     }
}
```

- javac SimpleApplet.java
- **appletviewer** SimpleApplet**.java**



➔This applet begins with two import statements.

➔The first imports the Abstract Window Toolkit (AWT) classes. Applets interact with the user through the AWT, not through the console-based I/O classes. The AWT contains support for a window-based, graphical interface.

➔The second import statement imports the applet package, which contains the class Applet. Every applet that you create must be a subclass of Applet.

➔Our class SimpleApplet must be declared as public, because it will be accessed by code that is outside the program.

➔ paint( ), is defined by the AWT Component class. Applets do not need a main( ) method. whenever the applet must redraw its output, paint( ) is called.

➔The paint( ) method has one parameter of type Graphics.

➔drawString( ), which is a member of the Graphics class. This method outputs a string beginning at the specified X,Y location.

Que. How Applets differ from Applications.

Ans.

Although both the applets and stand-alone applications are java programs, there are significant difference between them.

Applets are not full-featured application programs. They are usually written to accomplish a small task or a component of a task.

Since they are usually designed for use on the Internet, they impose certain limitations and restrictions in their design.

- Applets do not use **main( )** method for initiating the execution of the code. Applets, when loaded, automatically call certain methods of Applet class to start and execute the applet code.

- Unlike stand-alone applications, applets cannot be run independently. They are run from inside a web page using a special feature known as HTML tag.
- Applets cannot read from or write to the files in the local computer.
- Applets cannot communicate with other servers on the network.
- Applets cannot run any program from the local computer.
- Applets are restricted from using libraries from other languages such as C or C++. (Remember, Java languages supports this feature through native methods)

All these restrictions and limitations are placed in the interest of security of system. These restrictions ensure that an applet cannot do any damage to the local system.

Que. Explain Applet skeleton.
Ans.
All but the most 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.

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:

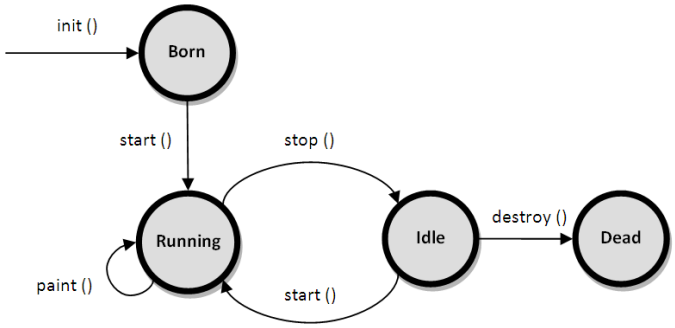
```
// An Applet skeleton.

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

/* <applet code="AppletSkel" width=300 height=100> </applet>*/

public class AppletSkel extends Applet
{
   public void init() { // initialization
   } /* Called first.*/

   public void start() { // start or resume execution
   } /* Called second, after init(). Also called whenever the applet is restarted. */

   public void stop() { // suspends execution
   } /* Called when the applet is stopped. */

   public void destroy() { // perform shutdown activities
   } /* Called when applet is terminated. This is the last method executed. */

   public void paint(Graphics g) { // redisplay contents of window
```

```
    } /* Called when an applet's window must be restored. */
}
```

Que. Explain Lifecycle of an applet.
Ans.
➔ When an applet is loaded, it undergoes a series of changes in its state as shown in fig.

➔ The applet states include :

- Born or Initialization state
- Running state
- Idle state or stopped state
- Dead or destroyed state



➔ **Initialization State :**
Applet enters the initialization state when it is first loaded. This is achieved by calling the **init()** method of Applet class. The applet is born. At this stage, we may do the following, if required.

- Create objects needed by the applet
- Set up initial values
- Load images or fonts
- Set up colors

```
public void init()
{
    ….
}
```

Running State :
Applet enters the running state when the system calls the **start()** method of Applet class. This occurs automatically after the applet is initialized. Starting can occur if the applet is already in 'stopped' state. For example, we may leave the Web page containing the applet temporarily to another page and return back to the page. This again starts the applet running. Note that, unlike init() method, the start() method may be **called more than once**. We may override the start() method to create a thread to control the applet.

```
public void start()
{
    ……
}
```

Display using paint( ) in Running State :
Applet moves to the display state whenever it has to perform some output operations on the screen. This happens immediately after the applet enters into the running state. The paint() method is called to accomplish this task. Almost every applet will have a paint() method.

It is to be noted that the **display state is not considered as a part of applet's life cycle**. In fact, the **paint()** method is defined in the Applet class. It is inherited from the **Component class**, a super class of Applet.

```
public void paint ( Graphics g )
{
     ….
 }
```

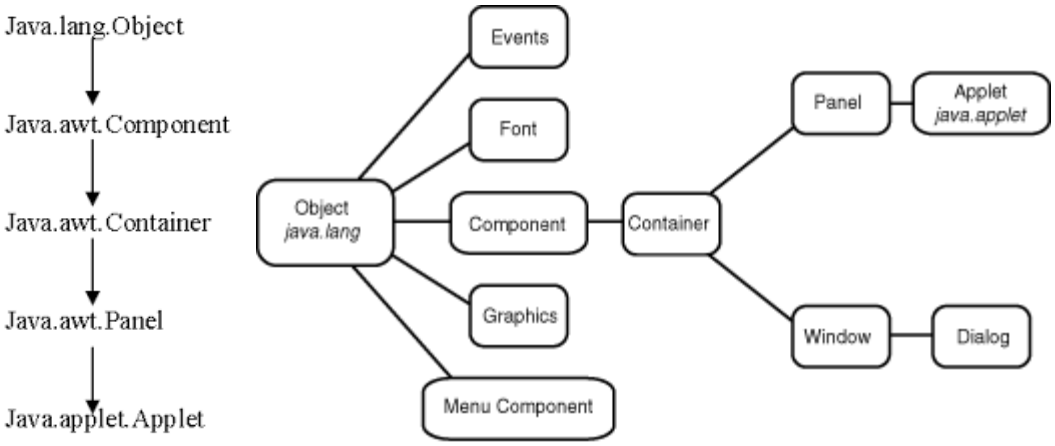
Idle or Stopped State :
An applet becomes idle when it is stopped from running. Stopping occurs automatically when we leave the page containing the currently running applet. We can also do so by calling the stop() method explicitly. If we use a thread to run the applet, then we must use stop() method to **terminate the thread**. We can achieve this by overriding the stop() method.

```
public void stop()
{
   ….
}
```

Dead State :
An applet is said to be dead when it is removed from memory. This occurs automatically by invoking the destroy() method when we quit the browser. Like initialization, destroying stage occurs **only once** in the applet's life cycle. If the applet has created any resources, like threads, we may override the destroy() method **to clean up** these **resources**. (Prof. Viral S. Patel)

```
public void destroy()
{
  ….
}
```

Que. **Explain the class hierarchy of Applet class.**
Ans. Applet class is available in **java.applet** package. The java.applet package also have AppletContext, AppletStub and AudioClip interfaces. Applet class have init( ), start( ), stop( ), destroy( ) methods. Applet class is subclass of **Panel class**.
➔Panel class is belong to **java.awt** package. Panel class is subclass of **Container class.**
(Window class is also subclass of Container class. Frame and Dialog are the subclasses of this Window class.) (Prof. Viral S. Patel)
➔**Container class** is subclass of Component which is also in same package **java.awt**. Button, Label, Canvas etc. are also subclasses of Component class.
➔**Component class** is subclass of Object. This Component class is belong to **java.awt** package. paint(Graphics g) method which we use in Applet is actually method of Component class.
➔Graphics, Font, Color, Events classes which we are used in Applet are subclass of **Object class**. Object class is belong to **java.lang** package.



Que. **How to pass parameters to Applet ?**
Ans. We can pass user-defined parameters to applet using **<param…>** tags.
Each <param…> tag has a name attribute and value attribute.
Inside the applet code, the applet can refer to that parameter by name to find its value. This is done using the **getParameter( )** method, which takes one string argument representing the name of the parameter and returns a string containing the value of that parameter.

Example : (Asst. Prof. Viral S. Patel)

```
import java.awt.*;
import java.applet.*;
/*
<applet code="MyApplet" width=800 height=600>
<param name="E_NAME" value=VIRAL>
<param name="E_AGE" value=30>
</applet>
*/
public class MyApplet extends Applet
{
  String name;
  int age;

  public void start()
  {
    name = getParameter("E_NAME");
    age = Integer.parseInt(getParameter("E_AGE"));
  }

  public void paint(Graphics g)
  {
    g.drawString(name + " " + age,20,20);
  }
}
```