Unit-5

Applets:-

1. **Concepts of Applets**
2. **Difference between applets and applications**
3. **Life cycle of an applet**
4. **Types of Applets**
5. **Creating applets**
6. **Passing parameters to applets**

1.Concepts of Applets

Defination:-

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side. An applet is embedded in an HTML page using the APPLET or OBJECT tag and hosted on a web server. Applets are used to make the website more dynamic and entertaining.

Important points :

1. Applets are not stand-alone programs. Instead, they run within either a web browser or an applet viewer. JDK provides a standard applet viewer tool called applet viewer.
2. Output of an applet window is not performed by *System.out.println()*. Rather it is handled with various AWT methods, such as *drawString()*.
3. An applet is a Java class that extends the java.applet.Applet class.
4. A main() method is not invoked on an applet, and an applet class will not define main().
5. Applets are designed to be embedded within an HTML page.
6. When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine.
7. 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.
8. The JVM on the user's machine creates an instance of the applet class and invokes various methods during the applet's lifetime.
9. 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.
10. Other classes that the applet needs can be downloaded in a single Java Archive (JAR) file.

Why we need Applet?

1. When we need to include something dynamic to include in our web page.
2. When we require some flash output like applet to produce some sound, animations or some special effects when displaying some certain pages.
3. When we want to create a program and it to make available on internet so that it could be used by others.

Advantage of Applet

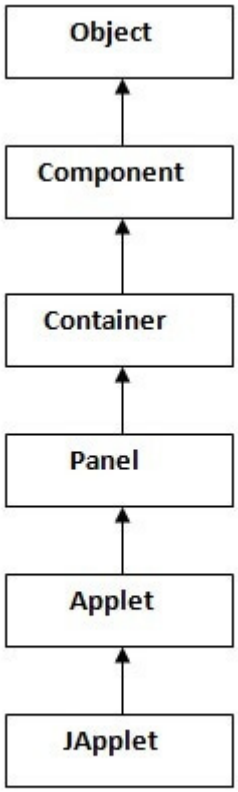
There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

- Plugin is required at client browser to execute applet.

Hierarchy of Applet



As displayed in the above diagram, Applet class extends Panel. Panel class extends Container which is the subclass of Component.

2.Difference between applets and applications

Parameters	Java Application	Java Applet
Definition	Applications are just like a Java program that can be executed independently without using the web browser.	Applets are small Java programs that are designed to be included with the HTML web document. They require a Java-enabled web browser for execution.
main () method	The application program requires a main() method for its execution.	The applet does not require the main() method for its execution instead init() method is required.
Compilation	The “javac” command is used to compile application programs, which are then executed using the “java” command.	Applet programs are compiled with the “javac” command and run using either the “appletviewer” command or the web browser.
File access	Java application programs have full access to the local file system and network.	Applets don’t have local disk and network access.
Access level	Applications can access all kinds of resources available on the system.	Applets can only access browser-specific services. They don’t have access to the local system.
Installation	First and foremost, the installation of a Java application on the local computer is required.	The Java applet does not need to be installed beforehand.
Execution	Applications can execute the programs from the local system.	Applets cannot execute programs from the local machine.
Program	An application program is needed to perform some tasks directly for the user.	An applet program is needed to perform small tasks or part of them.
Run	It cannot run on its own; it needs JRE to execute.	It cannot start on its own, but it can be executed using a Java-enabled web browser.

Parameters	Java Application	Java Applet
Connection with servers	Connectivity with other servers is possible.	It is unable to connect to other servers.
Read and Write Operation	It supports the reading and writing of files on the local computer.	It does not support the reading and writing of files on the local computer.
Security	Application can access the system's data and resources without any security limitations.	Executed in a more restricted environment with tighter security. They can only use services that are exclusive to their browser.
Restrictions	Java applications are self-contained and require no additional security because they are trusted.	Applet programs cannot run on their own, necessitating the maximum level of security.

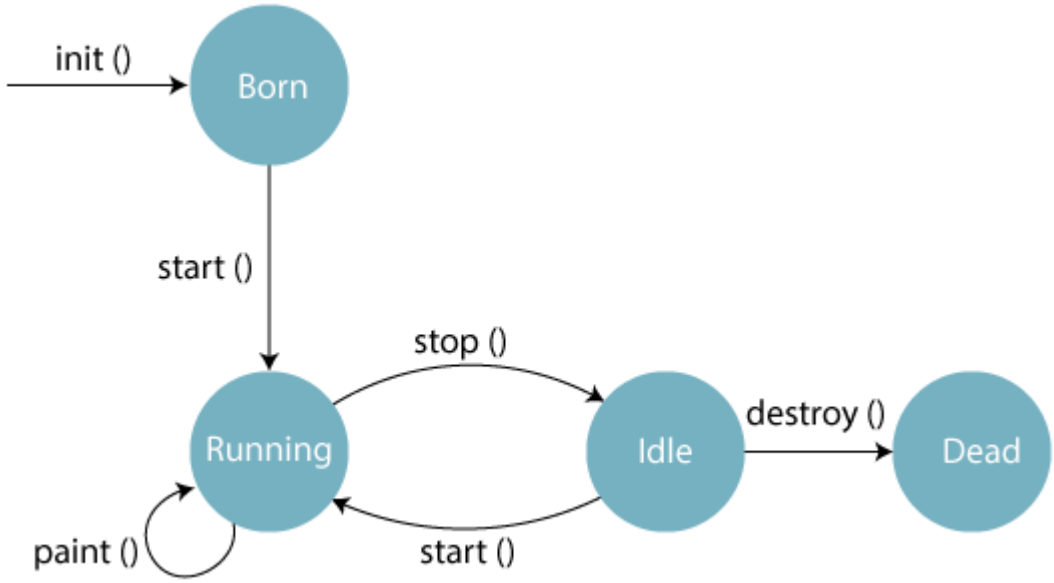
3.Life cycle of an applet

In Java, an applet is a special type of program embedded in the web page to generate dynamic content. Applet is a class in Java.

The applet life cycle can be defined as the process of how the object is created, started, stopped, and destroyed during the entire execution of its application. It basically has five core methods namely init(), start(), stop(), paint() and destroy().These methods are invoked by the browser to execute.

Along with the browser, the applet also works on the client side, thus having less processing time.

Methods of Applet Life Cycle

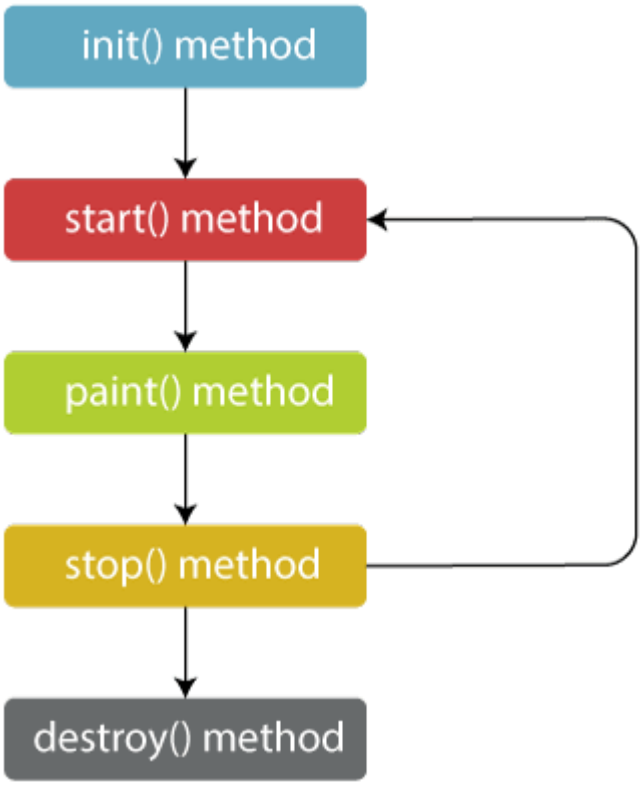


There are five methods of an applet life cycle, and they are:

- **init():** The init() method is the first method to run that initializes the applet. It can be invoked only once at the time of initialization. The web browser creates the initialized objects, i.e., the web browser (after checking the security settings) runs the init() method within the applet.
- **start():** The start() method contains the actual code of the applet and starts the applet. It is invoked immediately after the init() method is invoked. Every time the browser is loaded or refreshed, the start() method is invoked. It is also invoked whenever the applet is maximized, restored, or moving from one tab to another in the browser. It is in an inactive state until the init() method is invoked.
- **stop():** The stop() method stops the execution of the applet. The stop () method is invoked whenever the applet is stopped, minimized, or moving from one tab to another in the browser, the stop() method is invoked. When we go back to that page, the start() method is invoked again.
- **destroy():** The destroy() method destroys the applet after its work is done. It is invoked when the applet window is closed or when the tab containing the webpage is closed. It removes the applet object from memory and is executed only once. We cannot start the applet once it is destroyed.
- **paint():** The paint() method belongs to the Graphics class in Java. It is used to draw shapes like circle, square, trapezium, etc., in the applet. It is executed after the start() method and when the browser or applet windows are resized.

Flow of Applet Life Cycle:

These methods are invoked by the browser automatically. There is no need to call them explicitly.



Syntax of entire Applet Life Cycle in Java

```
1. class TestAppletLifeCycle extends Applet {
2. public void init() {
3. // initialized objects
4. }
5. public void start() {
6. // code to start the applet
7. }
8. public void paint(Graphics graphics) {
9. // draw the shapes
10.}
11.public void stop() {
12.// code to stop the applet
13.}
14.public void destroy() {
15.// code to destroy the applet
16.}
17.}
```

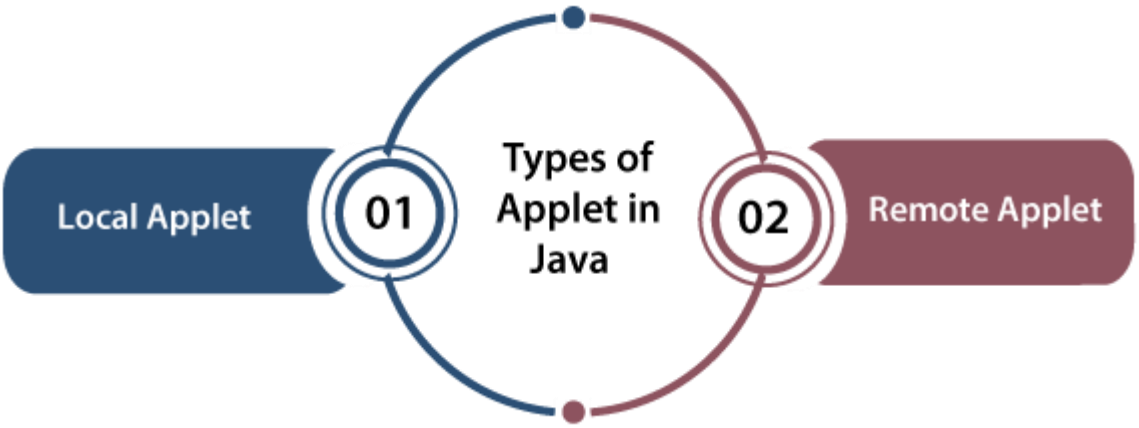
Who is responsible to manage the life cycle of an applet?

Java Plug-in software.

4.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.

1. **Local Applet**
2. **Remote Applet**



Let's understand both types of Applet one by one:

Local Applet

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

```
<applet
   codebase = "G:\anu"
   code = "FaceApplet.class"
   width = 120
   height = 120>
</applet>
```

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.

FaceApplet.java

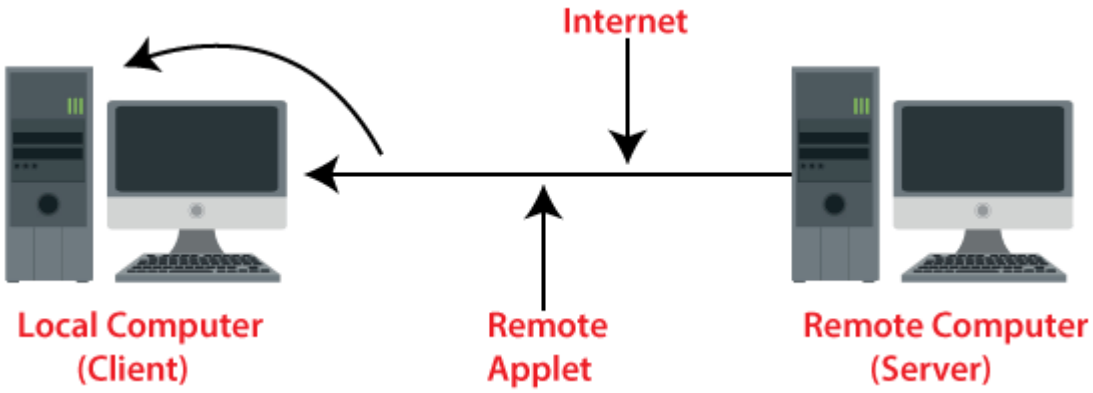
```
//Import packages and classes
import java.applet.*;
import java.awt.*;
//Creating FaceApplet class that extends Applet
public class FaceApplet extends Applet
{
   //paint() method starts
   public void paint(Graphics g){
      //Creating graphical object
      g.setColor(Color.red);
      g.drawString("Welcome", 50, 50);

   }
}
```

Execute the above code by using the following commands:

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).



Specifying Remote applet

1. **<applet**
2. codebase = "http://www.myconnect.com/applets/"
3. code = "FaceApplet.class"

4. width = 120
5. height =120>
6. **</applet>**

Difference Between Local Applet and Remote Applet

The following table describes the key differences between Local applet and Remote applet.

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.

5.Creating applets

Creating Hello Geeks applet :

There are two ways to run an applet

1. **By using Web Browser**
2. **By using Applet Viewer**

1.Executing the applet within a Java-compatible Web browser:
Suppose we have a GfgApplet.java file in which we have our java code.

// Java program to run the applet

```
// using the web browser

import java.awt.*;

import java.applet.*;

public class GfgApplet extends Applet

{

    public void paint(Graphics g)

    {

        g.drawString("Hello Geeks,Welcome to GeeksforGeeks",20,20);

    }

}
```

- Compiling: javac GfgApplet.java
- Create an Html file and embed the Applet tag in the HTML file.

To run an applet in a web browser, we must create an HTML text file with a tag that loads the applet. For this, we may use the APPLET or OBJECT tags. Here is the HTML file that runs HelloWorld with APPLET:

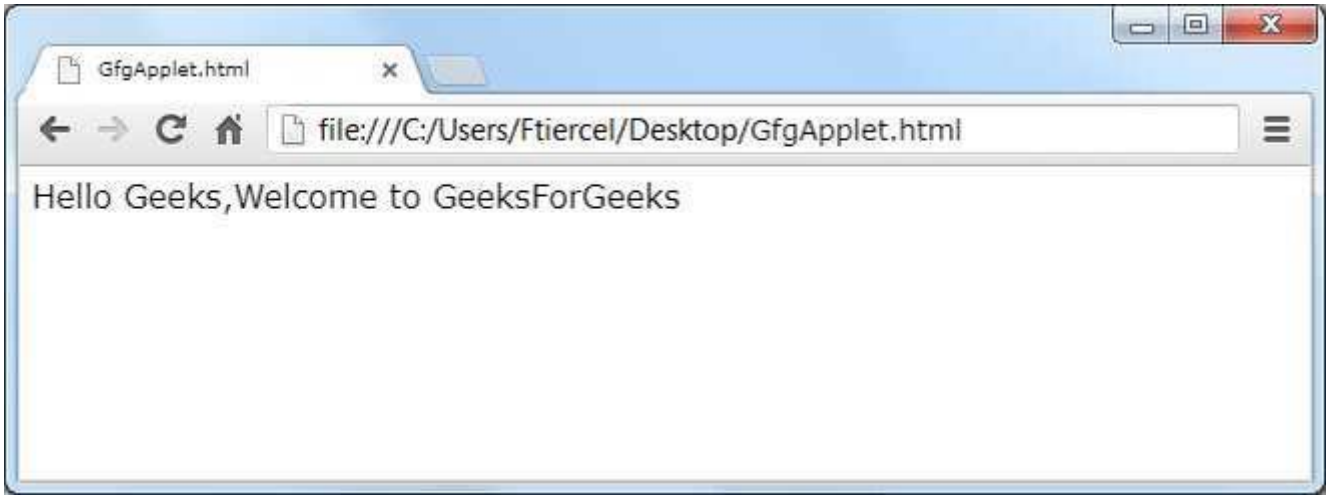
Attributes in applet tag:

- **Code:** It specifies the name of the applet class to load into the browser.
- **Width:** It specifies the width of an applet.
- **Height:** It sets the height of an applet.

GfgApplet.html

```
<html>
<body>
<applet code="GfgApplet.class" width=300 height=100></applet>
</body>
</html>
```

Output on browser:



The applet GfgApplet.class is loaded into the browser when you access GfgApplet.html.

To load applet programs, the browser must have java enabled.

Double click on GfgApplet.html

This **won't work** on browser as we don't have the **proper plugins.**

You can run directly applet viewer using appletviewer tool

- **Javac GfgApplet.java**
- **Appletviewer GfgApplet.html**

2. Using an Applet Viewer to run the applet:
It is a java application that allows you to view applets. It's similar to a mini-browser that will enable you to see how an applet might appear in a browser. It recognizes the APPLET tag and uses it during the creation process. The APPLET tag should be written in the source code file, with comments around it.

- Write HTML APPLET tag in comments in the source file.
- Compile the applet source code using javac.
- Use applet viewer ClassName.class to view the applet.

```
// Java program to run the applet

// using the applet viewer

import java.awt.*;
```

```
import java.applet.*;

public class GfgApplet extends Applet

{

    public void paint(Graphics g)

    {

        g.drawString(“Hello Geeks,Welcome to GeeksforGeeks”,20,20);

    }

   }

/*

<applet code="GfgApplet" width=300 height=100>

</applet>

*/
```

To use the applet viewer utility to run the applet, type the following at the command prompt:

c:\>javac GfgApplet.java
c:\>appletviewer GfgApplet.java
Output:

6.Passing parameters to applets

The APPLET tag in HTML allows you to pass parameters to your applet. To retrieve a parameter, use the getParameter( ) method. It returns the value of the specified parameter in the form of a String object.

Param Tag

The <param> tag is a sub tag of the <applet> tag. The <param> tag contains two attributes: *name* and *value* which are used to specify the name of the parameter and the value of the parameter respectively. For example, the param tags for passing name and age parameters looks as shown below:

```
<param name=”name” value=”Ramesh” />
<param name=”age” value=”25″ />
```

Now, these two parameters can be accessed in the applet program using the *getParameter()* method of the *Applet* class.

getParameter() method:-

The *getParameter()* method of the *Applet* class can be used to retrieve the parameters passed from the HTML page. The syntax of *getParameter()* method is as follows:

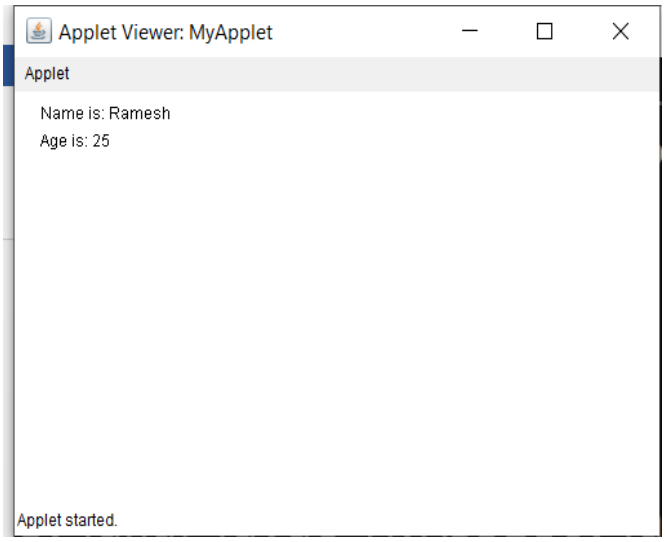
```
String getParameter(String param-name)
```

Let’s look at a sample program which demonstrates the <param> HTML tag and the *getParameter()* method:

```
import java.awt.*;
import java.applet.*;
public class MyApplet extends Applet
{
	String n;
	String a;
	public void init()
	{
		n = getParameter("name");
		a = getParameter("age");
	}
	public void paint(Graphics g)
	{
		g.drawString("Name is: " + n, 20, 20);
		g.drawString("Age is: " + a, 20, 40);
	}
```

```
}
/*
        <applet code="MyApplet" height="300" width="500">
                <param name="name" value="Ramesh" />
                <param name="age" value="25" />
        </applet>
*/
```

OutPut:-



Next Topic:-

The HTML APPLET Tag:

< APPLET

[CODEBASE = codebaseURL]

CODE = appletFile

[ALT = alternateText]

[NAME = appletInstanceName]

WIDTH = pixels

HEIGHT = pixels

[ALIGN = alignment]

[VSPACE = pixels]

[HSPACE = pixels]

>

[< PARAM NAME = AttributeName VALUE = AttributeValue>]

[< PARAM NAME = AttributeName2 VALUE = AttributeValue>]

. . .

[HTML Displayed in the absence of Java]

</APPLET>

Let's take a look at each part now.

CODEBASE: CODEBASE is an optional attribute that specifies the base URL of the applet code, which is the directory that will be searched for the applet's executable class file (specified by the CODE tag).

CODE :CODE is a required attribute that gives the name of the file containing your applet's compiled .class file. This file is relative to the code base URL of the applet, which is the directory that the HTML file was in or the directory indicated by CODEBASE if set.

ALT :The ALT tag is an optional attribute used to specify a short text message that should be displayed if the browser understands the APPLET tag but can't currently run Java applets.

NAME: NAME is an optional attribute used to specify a name for the applet instance. Applets must be named in order for other applets on the same page to find them by name and communicate with them.

WIDTH AND HEIGHT :WIDTH and HEIGHT are required attributes that give the size (in pixels) of the applet display area.

ALIGN: ALIGN is an optional attribute that specifies the alignment of the applet.The possible values:LEFT, RIGHT, TOP, BOTTOM, MIDDLE, BASELINE, TEXTTOP, ABSMIDDLE,and ABSBOTTOM.

VSPACE AND HSPACE :These attributes are optional. VSPACE specifies the space, in pixels, above and below the applet. HSPACE specifies the space, in pixels, on each side of the applet.

PARAM NAME AND VALUE: The PARAM tag allows you to specify applet specific arguments in an HTML page. Applets access their attributes with the getParameter( ) method.

What is the use of repaint ()?

repaint(): This method cannot be overridden. It controls the update() -> paint() cycle. We can call this method to get a component to repaint itself. the repaint() method is invoked to refresh the viewing area i.e. you call it when you have new things to display.