8.1 Fundamental concepts of applets:

- An applet is a Java program that can be embedded into a web page but they 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.
- It runs inside the web browser and works at client side.
- Applets are designed to be embedded within the HTML page, thereby, providing functionalities that a simple HTML page cannot provide.
- Applets are used to make the web site more dynamic and entertaining.
- Java Applets are usually used to add small, interactive components or enhancements to a webpage.
- These may consist of buttons, scrolling text, or stock tickers, but they can also be used to display larger programs like word processors or games. Java's incredible portability (the ability to run on many different operating systems and browsers) is what makes it ideal for use on the World Wide Web.
- All applets are java sub-classes (either directly or indirectly) of *java.applet.applet* class.
- Applet is not used nowadays it is out dated technology

8.2 Simple Applet and Applet & Application.

- An applet is a small application that performs specific operations or functions. It is a utility program which loaded from the web server and executed by the web browser.
- Java has transformed the manner the Internet users retrieve and use documents on the world wide network.
- Applets have enabled to construct and use completely interactive multimedia Web documents.
- A web page can include a java applet which support several applications such as executing arithmetic operations, the creation of animation, graphics displaying, playing interactive games.

There are two ways an applet can integrate into web pages.

1. **Local applet**:
   - ✓ In which we can write our own applets and integrate them into web pages.

- ✓ These types of applets evolved locally and kept in a local system.

2. Remote applet

- ✓ These types of applets are developed externally and stored on a remote computer to the Internet or (we can download an applet from a remote computer system and then embed it into a web page.)

Application

- ❖ An Application is a program which runs on an underlying operating system.
- ❖ Application are generic in a sense and designed to perform a specific task directly for the user.
- ❖ Application can run with or without GUI.
- ❖ The application programs like spreadsheets, word processors, web browsers and compilers – describe the manners in which computer resources are utilized to resolve users' computing issues.

Java Application:

- ❖ A Java application is a standalone java program that runs on client or server.
- ❖ Java application is a program that runs on itself as and when the main () method of any specific class is executed.
- ❖ A java application has a functionality that benefits user or some other java application.

Applet Vs Application:

Java Application	Java Applet
An application is a standalone Java program which can be run independently on client/server without the need of a web browser.	An applet is a form of Java program which is embedded with HTML page and loaded by a web server to be run on a web browser.
The execution of the program starts from the main () method.	There is no requirement of main () method for the execution of the program.
Application can access local disk files/ folders and network system.	Applet doesn't have access to the local network files and folders.

It doesn't require any Graphical User Interface (GUI).	It must run within GUI.
It is a trusted application and doesn't require much security.	It requires high-security constraints as applets are untrusted.
It requires Java Runtime Environment (JRE) for its successful execution.	It requires a web browser like Chrome, Firefox, etc. for its successful execution.
It is explicitly run and installed on a local system.An applet doesn't have access to local files and so it cannot perform read and write operations on files stored on local disk.	It doesn't require any explicit installation to be done.
An application can perform read and write operations on files stored on the local disk.	An applet doesn't have access to local files and so cannot perform read and write operations on files stored on local disk.

`public class ApplicationExample {`
 	
`public static void main(String args[ ])`
 	
`{`
 	
`System.out.println("Hello, I am Application");`
 	
`}`
 	
`}`
	`import java.awt.*;`
	`import java.applet.*;`
	
	`public class AppletExample extends Applet {`
	`public void init() { }`
	`public void start() { }`
	`public void stop() {}`
	`public void destroy() {}`
	`public void paint(Graphics g) {}`
	`}`

Life Cycle of Applet:

- As shown in the below diagram, the life cycle of an applet starts with *init()*method and ends with *destroy()* method.
- The methods to execute only once in the applet life cycle are *init()* and *destroy()*.
- Other methods execute multiple times.
- Java Plug-in software are responsible to manage the life cycle of an applet

There are two ways to run an applet

1. By html file.
2. By applet Viewer tool (for testing purpose).

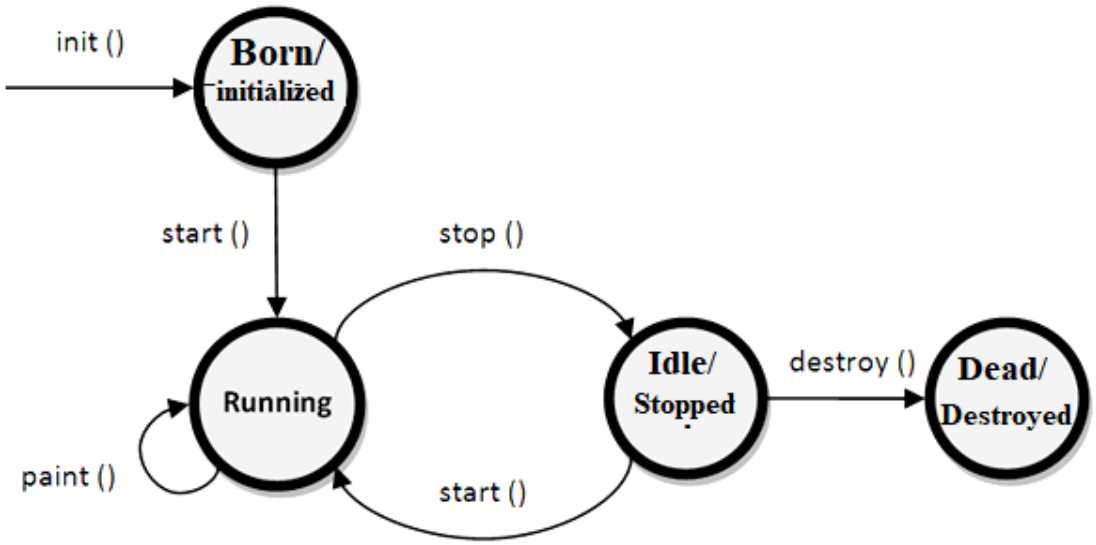

Figure: Life Cycle of an Applet.

For creating any applet ***java.applet.Applet*** class must be inherited and it provides 4 life cycle methods [int(), start(), stop(), destroy()] of applet when ***java.awt.Component*** class provides 1 life cycle methods for an applet.

1. Applet is initialized.
2. Applet is started.
3. Applet is painted.
4. Applet is stopped.
5. Applet is destroyed.

- ❖ When the user opens Applet Life, an HTML page containing the applet life cycle.
- ❖ Java in an app1et viewer, an instance for each of the applet classes is created using the no-argument constructor, and its **init()** method called.
- ❖ As a result, the message**" init() called**" is displayed in the command window.

- After execution of the code in **the init()** method, the program control returns to the applet viewer which then calls the applet's **start ()** method.
- As a result, the message "**start() called**" is displayed.
- Next, the applet viewer calls the applet's **paint()** method as a result of which the message "**paint() called**" displayed.
- Now when the user minimizes the applet window, the **stop()** method is called, and on restoring the window the **start()** method is called again.
- When the user exits the appletviewer, the **destroy()** method called that displays the message "**destroy() called"** in the command window.

Role of Life Cycle method

1. **init():**

- ✓ The init() method is the first method to execute when the applet is executed.
- ✓ Variable declaration and initialization operations are performed in this method.

2. **start():**

   - ✓ The start() method contains the actual code of the applet that should run.
   - ✓ The start() method executes immediately after the *init()* method.
   - ✓ It also executes whenever the applet is restored, maximized or moving from one tab to another tab in the browser.

3. **stop():**
   - ✓ The stop() method stops the execution of the applet.
   - ✓ The stop() method executes when the applet is minimized or when moving from one tab to another in the browser.
4. **destroy():**
   - ✓ The destroy() method executes when the applet window is closed or when the tab containing the webpage is closed.
   - ✓ *stop()* method executes just before when destroy() method is invoked.
   - ✓ The destroy() method removes the applet object from memory.
5. **paint():**
   - ✓ The paint() method is used to redraw the output on the applet display area.
   - ✓ The paint() method executes after the execution of *start()* method and whenever the applet or browser is resized.

- ✓ **public void paint(Graphics g)**provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

The method execution sequence when an applet is executed is:

1. init()
2. start()
3. paint()

The method execution sequence when an applet is closed is:

1. stop()
2. destroy()

Example program to demonstrate the life cycle of an applet

```
import java.awt.*;
import java.applet.*;
public class MyAppletLife extends Applet
{
    public void init()
    {
        System.out.println("Applet initialized");
    }
    public void start()
    {
        System.out.println("Applet execution started");
    }
    public void stop()
    {
        System.out.println("Applet execution stopped");
    }
    public void paint(Graphics g)
    {
        System.out.println("Painting...");
    }
    public void destroy()
    {
        System.out.println("Applet destroyed");
    }
}
```

Simple example of Applet by html file:

1. To execute the applet by html file, create an applet and compile it.
2. After that create an html file and place the applet code in html file.
3. Then click the html file
4. To execute the applet by applet viewer tool, write in command prompt:

```
                                        //AppletExample.java
import java.applet.Applet;
import java.awt.Graphics;

public class IamApplet extends Applet // IamApplet class extends Applet

{

   public void paint(Graphics g)
   {
      g.drawString ("Hello from am Applet Example", 50, 50); // string, coordinates
   }

}
```

Here, this code file is saved as “AppletExample.java” and after compilation this becomes “AppletExample.class”

Now, embedded this code inside HTML write following code.

```
                              //ItsMyAppletExample.html
<html>
<title> My Applet Example</title>
<applet code="AppletExample.class" width = 500 height = 500>
</applet>
</html>
```

Passing Parameter to applets:

<PARAM… > tag is used to pass the parameter value from HTML file to Applet code.

```
//myHtmlCode.html
<html>
<title> Test Parameter</title>
<applet code = myApplet.class width = 500 height=500>
<PARAM Name ="pname" value="I am example of passing Parameter to applet">
</applet>
</html>
```

```
//myApplet.java
import java.applet.*;
import java.awt.*;
public class myApplet extends applet
{
   string str = null;
   public void init()
   {
      str= getParmeter("pname");
   }
   public void paint(Graphics g)
   {
      g.drawString(str,100,100);
      }
}
```

In above example, if you want to initialize any value in java applet then you have to use ***public void init()*** method so value of ***string str*** is initialize to take from ***html*** file.

8.3 Applet Architecture:

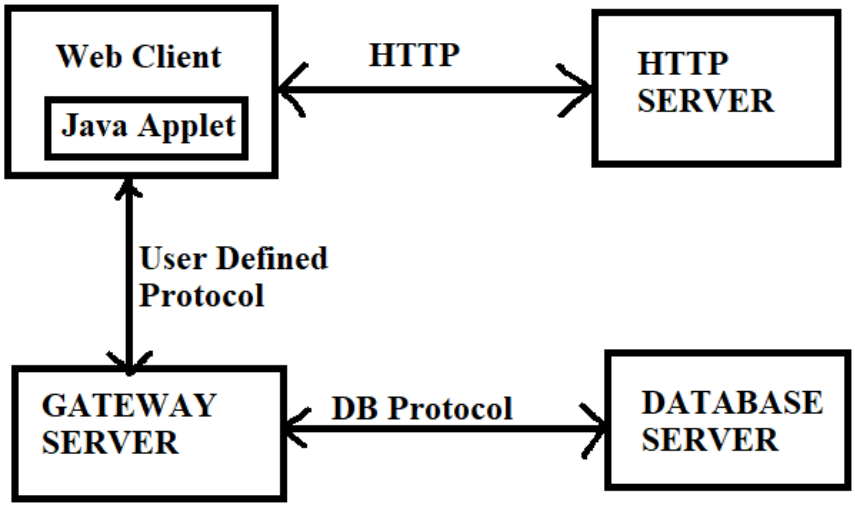

Figure: Architecture of Java Applet

We know that java provides console based programming language environment and window based programming environment.

- An applet is a window based programming environment. So applet architecture is different than console base program.
- Java applets are essentially java window programs that can run within a web page. Applet programs are java classes that extend that java.applet.
- Applet class and are enabled by reference with HTML page. You can observed that when applet are combined with HTML, that can make an interface more dynamic and powerful than with HTML alone.
- While some Applet do not more than scroll text or play movements, but by incorporating theses basic features in webpage you can make them dynamic.
- These dynamic web page can be used in an enterprise application to view or manipulate data coming from some source on the server.
    - ✓ The Applet and there class files are distribute through standard HTTP request and therefore can be sent across firewall with the web page data.
    - ✓ Applet code is referenced automatically each time the user revisit the hosting website.
    - ✓ Therefore keeps full application up to date on each client desktop on which it is running.

Advantage and Disadvantage of Applet:

Advantages:

1. it is simple to make it work on windows, Linux and MAC OS i.e. cross platform
2. The same applet can work on all installed versions of JAVA at the same time
3. It can move the work from server to client
4. Secured:- An un-trusted applet has no access to the local machine and can only access the server it came from
5. Java applets are fast
6. They increase interactivity for users
7. Java has constituted security, by this feature of java; an applet doesn't produce any harmful activity to your computer.
8. Database integration is another important advantage of applets.
9. They can pay out different type of action on client-side machine like-
    - Playing sound
    - Viewing images
    - Get user input
    - Get mouse clicks
    - Get user keystrokes

Disadvantage:

1. It requires the java plug-in
2. Some applets require a specific JRE
3. Some browsers especially mobile browsers which are running on Apple IOS or Android don't support applets run.
4. Applets don't access client-side resources, like Files, Operating system
5. Applet can only extract information about client-machine is its name, java version, OS, version etc.

Applet Security Policies:

Due to security reasons, the following restrictions are imposed on Java applets:

1. An applet cannot load libraries or define native methods.
2. An applet cannot ordinarily read or write files on the execution host.
3. An applet cannot read certain system properties.
4. An applet cannot make network connections except to the host that it came from.
5. An applet cannot start any program on the host that's executing it.

How to convert applet into application? (vvimp)

Following applet changes the background of the window as per the button clicked.

Applet:

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

public class ClickButton extends Applet implements ActionListener
{
  Button rb, gb, bb;

  public void init()
  {
    setLayout(new FlowLayout());// for applet, FlowLayout is the defau
lt also

    rb = new Button("Red");
    gb = new Button("Green");
    bb = new Button("Blue");

    rb.addActionListener(this);
    gb.addActionListener(this);
    bb.addActionListener(this);

    add(rb);
    add(gb);
    add(bb);   // no set methods
  }
```

```
  public  void actionPerformed(ActionEvent e)
  {
    String str = e.getActionCommand();

    if(str.equals("Red"))
      setBackground(Color.red);
    else if(str.equals("Green"))
      setBackground(Color.green);
    else if(str.equals("Blue"))
      setBackground(Color.blue);
  }
} // class closes, no main()
```

HTML file code to execute above applet

```
<applet code="ClickButton.class" width="500" height="500">
</applet>
```

For the conversion of Applet to Application do following:

1. Delete the statement *"**import java.applet**" package*
2. Replace ***extends*** Applet with ***Frame***
3. Replace the ***init()*** method with constructor
4. Check the layout (for Applet, the default layout is FlowLayout and for Frame, it is BorderLayout)
5. Add setXXX() methods like setSize() etc.
6. Add the ***main()*** method
7. HTML is not required (delete it)

Application obtained after converting the Applet to Application.

```
import java.awt.*;
import java.awt.event.*;

public class ClickButton extends Frame implements ActionListener
{
  Button rb, gb, bb;

  public ClickButton()
  {
    setLayout(new FlowLayout());
```

```
    rb = new Button("Red");
    gb = new Button("Green");
    bb = new Button("Blue");

    rb.addActionListener(this);
    gb.addActionListener(this);
    bb.addActionListener(this);

    add(rb);
    add(gb);
    add(bb);

    setTitle("Hello I am the Application after conversion from Applet"
);
    setSize(500, 500);
    setVisible(true);
  }
  public  void actionPerformed(ActionEvent e)
  {
    String str = e.getActionCommand();

    if(str.equals("Red"))
      setBackground(Color.red);
    else if(str.equals("Green"))
      setBackground(Color.green);
    else if(str.equals("Blue"))
       setBackground(Color.blue);
  }
  public static void main(String args[])
  {
    new ClickButton();
  }
}
```