



K. J. Somaiya College of Engineering, Mumbai-77

Batch: B1

Roll No.: 1711072

Experiment / assignment / tutorial No.10

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of the Staff In-charge with date

TITLE: Creating an applet.

AIM : Write a java applet program for displaying the factorial of a number when user enters a number.

Expected OUTCOME of Experiment:

CO4: Demonstrate programs on interface, exceptions, multithreading and applets.

Books/ Journals/ Websites referred:

- 1.Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” Tata McGraw-Hill.
 - 2.Grady Booch, Object Oriented Analysis and Design .
-

Pre Lab/ Prior Concepts:

Theory: Applet is deprecated in Java 8 and above because of security issues since Applets aren't secure. An applet is a small Java program that is embedded and ran in some other Java interpreter program such as Java technology-enabled browser, Sun's applet viewer program called appletviewer.

Creating SimpleApplet in java:

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

```
public class HelloWorldApplet extends Applet
```



K. J. Somaiya College of Engineering, Mumbai-77

```
{  
    public void paint (Graphics g)  
    {  
        g.drawString ("Hello World", 25, 50);  
    }  
}
```

First two import statements bring the classes into the scope of our applet class. Without those import statements, the Java compiler would not recognize the classes Applet and Graphics, which the applet class refers to.

Running Applet in java:

An applet may be invoked by embedding directives in an HTML file and viewing the file through an applet viewer or Java-enabled browser.

The <applet> tag is the basis for embedding an applet in an HTML file. Below is an example that invokes the "Hello, World" applet:

```
<html>  
<title>The HelloWorld Applet</title>  
<body>  
<hr>  
<applet code="HelloWorldApplet.class" width="320" height="120">  
<param Name="string" Value="Hello">  
<param name="audio" value="test.wav">  
If your browser was Java-enabled, a "Hello, World"  
message would appear here.  
  
</applet>  
<hr>  
</body>  
</html>
```

The code attribute of the <applet> tag is required. It specifies the Applet class to run. Width and height are also required to specify the initial size of the panel in which an applet runs. The applet directive must be closed with a </applet> tag.

If an applet takes parameters, values may be passed for the parameters by adding <param> tags between <applet> and </applet>. The browser ignores text and other tags between the applet tags.



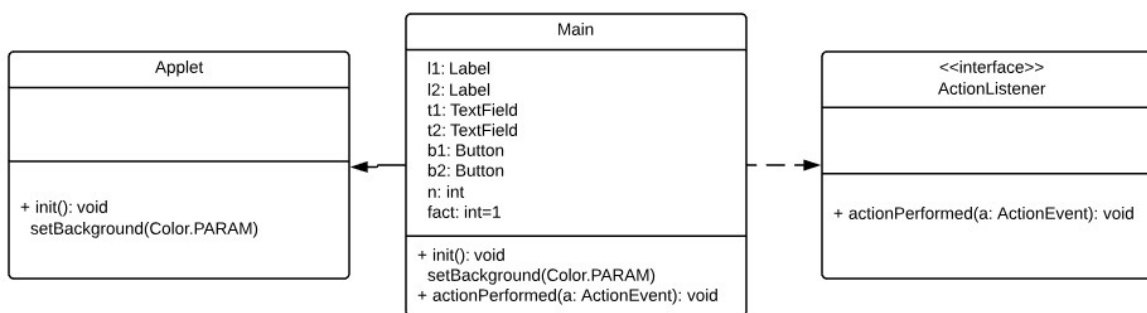
K. J. Somaiya College of Engineering, Mumbai-77

Non-Java-enabled browsers do not process `<applet>` and `</applet>`. Therefore, anything that appears between the tags, not related to the applet, is visible in non-Java-enabled browsers.

Attributes of Applet tag:

Attribute	Explanation	Example
Code	Name of class file	Code="applet0.class"
Width	Width of applet	Width=300
height	Height of applet	Height=60
Codebase	Applet's Directory	Codebase="/applets"
Alt	Alternate text if applet not available	Alt="menu applet"
name	Name of the applet	Name="appletExam"
Align (top,left,right,bottom)	Justify the applet with text	Align="right"

Class Diagram:



Implementation details:

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

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



K. J. Somaiya College of Engineering, Mumbai-77

```
public class applet extends Applet implements ActionListener
{
    Label l1,l2;
    TextField t1,t2;
    Button b1,b2;
    public void init()
    {
        setBackground(Color.RED);
        l1 = new Label("ENTER THE NUMBER: ");
        l2 = new Label("RESULT: ");
        t1 = new TextField(20);
        t2 = new TextField(20);
        t2.setEnabled(false);
        b1 = new Button("COMPUTE");
        b2 = new Button("CLEAR");
        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(b1);
        add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
        l2.setVisible(false);
        t2.setVisible(false);
        b2.setVisible(false);
    }

    public void actionPerformed(ActionEvent a)
    {
        int n = Integer.parseInt(t1.getText());
        int fact=1;
        if(a.getSource() == b1)
        {
            l2.setVisible(true);
            t2.setVisible(true);
            l1.setVisible(false);
            l2.setVisible(false);
            t1.setVisible(false);
            b2.setVisible(true);
            b1.setVisible(false);
            setBackground(Color.GREEN);
            if(n==0 || n==1)
            {
```



K. J. Somaiya College of Engineering, Mumbai-77

```
        fact=1;
    }
    else
    {
        for(int i = 1;i<=n;i++)
        {
            fact=fact*i;
        }
    }
    t2.setText(String.valueOf(fact));
}
else if(a.getSource()==b2){
    l1.setVisible(true);
    b1.setVisible(true);
    l2.setVisible(false);
    t1.setVisible(true);
    t2.setVisible(false);
    b2.setVisible(false);
    setBackground(Color.RED);
    t1.setText("");
    t2.setText("");
}
}
}
```

To verify my code, here's my repl link:

<https://repl.it/@ARGHYADEEPPDAS/Applet-OOPM>

Output GIF:

Attached along with this pdf. Please find the GIF.

Conclusion: The program ran successfully as we were able to compute factorial, hide components at will and also change background color.

Post Lab Descriptive Questions (Add questions from examination point view)

1. Explain Applet Life cycle.

Ans. The various states that an applet undergoes between its object creation and object removal (when the job is over) are known as **Applet Life Cycle**. Each state is



K. J. Somaiya College of Engineering, Mumbai-77

represented by a method. There exist 5 states represented by 5 methods. That is, in its life of execution, the applet exists (lives) in one of these 5 states.

These methods are known as "**callback methods**" as they are called automatically by the browser whenever required for the smooth execution of the applet. Programmer just writes the methods with some code but never calls.

Following are the methods.

1. `init()` method
2. `start()` method
3. `paint()` method
4. `stop()` method
5. `destroy()` method

These methods are known as Applet Life Cycle methods. These methods are defined in **java.applet.Applet** class except **paint()** method. The **paint()** method is defined in **java.awt.Component** class, an indirect super class of Applet.

Browser Responsibilities

The Applet Life Cycle methods are called as **callback methods** as they are called implicitly by the browser for the smooth execution of the applet. Browser should provide an environment known as **container** for the execution of the applet. Following are the responsibilities of the browser.

1. It should call the callback methods at appropriate times for the smooth execution of the applet.
2. It is responsible to maintain the Applet Life Cycle.
3. It should have the capability to communicate between applets, applet to JavaScript and HTML, applet to browser etc.

Description of Applet Life Cycle methods

Even though, the methods are called automatically by the browser, the programmer should know well when they are called and what he can do with the methods.

Following is the schematic representation of the methods.

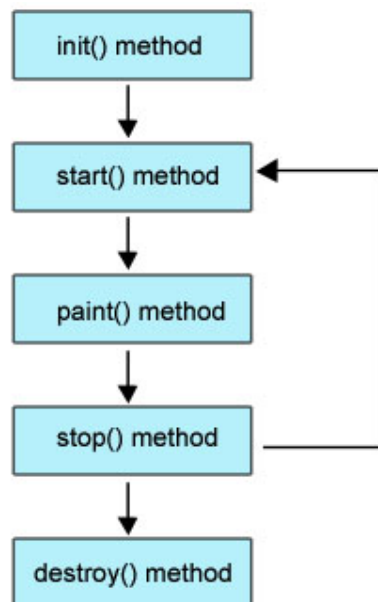


Figure: Life cycle of Applet

Brief Description of Life Cycle Methods

Following is the brief description of the above methods.

1. **init():** The applet's voyage starts here. In this method, the applet object is created by the browser. Because this method is called before all the other methods, programmer can utilize this method to instantiate objects, initialize variables, setting background and foreground colors in GUI etc.; the place of a constructor in an application. It is equivalent to **born state** of a thread.
2. **start():** In init() method, even though applet object is created, it is in **inactive** state. An inactive applet is not eligible for microprocessor time even though the microprocessor is idle. To make the applet active, the init() method calls start() method. In start() method, applet becomes active and thereby eligible for processor time.
3. **paint():** This method takes a **java.awt.Graphics** object as parameter. This class includes many methods of drawing necessary to draw on the applet window. This is the place where the programmer can write his code of what he expects from applet like animation etc. This is equivalent to **runnable state** of thread.
4. **stop():** In this method the applet becomes temporarily inactive. An applet can come any number of times into this method in its life cycle and can go back to the active state (paint() method) whenever would like. It is the best place to have cleanup code. It is equivalent to the **blocked state** of the thread.
5. **destroy():** This method is called just before an applet object is garbage collected. This is the end of the life cycle of applet. It is the best place to have cleanup code. It is equivalent to the **dead state** of the thread.



K. J. Somaiya College of Engineering, Mumbai-77

After knowing the methods, let us know when they are called by the browser.

- **init()** method is called at the time of starting the execution. This is called only once in the life cycle.
- **start()** method is called by the init() method. This method is called a number of times in the life cycle; whenever the applet is **deiconified**, to make the applet active.
- **paint()** method is called by the start() method. This is called number of times in the execution.
- **stop()** method is called whenever the applet window is **iconified** to inactivate the applet. This method is called number of times in the execution.
- **destroy()** method is called when the applet is closed. This method is called only once in the life cycle.

Observe, the init() and destroy() methods are called only once in the life cycle. But, start(), paint() and stop() methods are called a number of times.

2. Write the difference between Applets and Standalone Java applications.

BASIS FOR COMPARISON	APPLET	APPLICATION
Basic	It is small program uses another application program for its execution.	An application is the programs executed on the computer independently.
main() method	Do not use the main method	Uses the main method for execution
Execution	Cannot run independently require API's (Ex. Web API).	Can run alone but require JRE.
Installation	Prior installation is not needed	Requires prior explicit installation on



K. J. Somaiya College of Engineering, Mumbai-77

BASIS FOR COMPARISON	APPLET	APPLICATION
		the local computer.
Read and write operation	The files cannot be read and write on the local computer through applet.	Applications are capable of performing those operations to the files on the local computer.
Communication with other servers	Cannot communicate with other servers.	Communication with other servers is probably possible.
Restrictions	Applets cannot access files residing on the local computer.	Can access any data or file available on the system.
Security	Requires security for the system as they are untrusted.	No security concerns are there.

Date: 19/10/2018

Signature of faculty in-charge

Department of Computer Engineering