Applets Programming

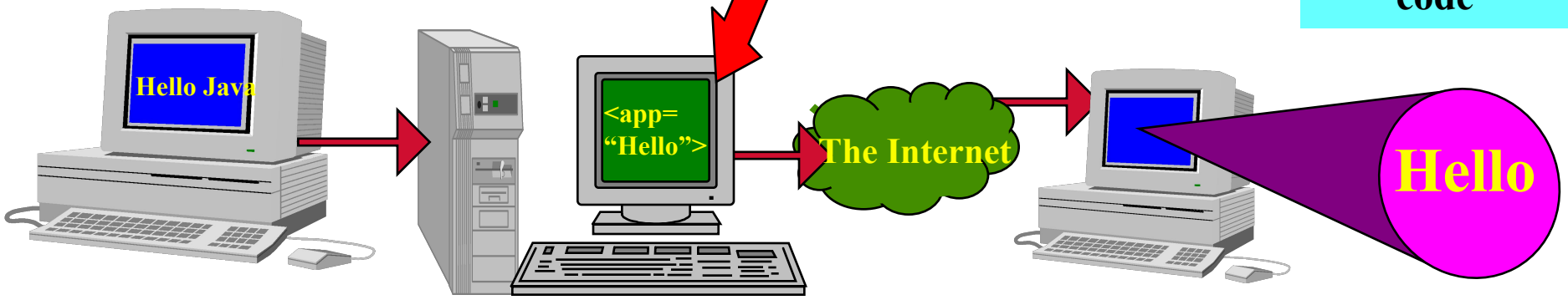
Enabling Application Delivery Via the Web

Introduction

- Applets are small Java programs that are embedded in Web pages.
- They can be transported over the Internet from one computer (web server) to another (client computers).
- They transform web into rich media and support the delivery of applications via the Internet.

Applet: Making Web Interactive and Application Delivery Media

1. APPLET Development "hello.java" AT SUN.COM
2. hello.class AT SUN'S WEB SERVER
3. Create Applet tag in HTML document
4. Accessing from Your Organisation
5. The browser creates a new window and a new thread and then runs the code



How Applets Differ from Applications

- Although both the Applets and stand-alone applications are Java programs, there are certain restrictions are imposed on Applets due to security concerns:
  - Applets don't use the main() method, but when they are load, automatically call certain methods (init, start, paint, stop, destroy).
  - They are embedded inside a web page and executed in browsers.
  - They cannot read from or write to the files on local computer.
  - They cannot communicate with other servers on the network.
  - They cannot run any programs from the local computer.
  - They are restricted from using libraries from other languages.
- The above restrictions ensures that an Applet cannot do any damage to the local system.

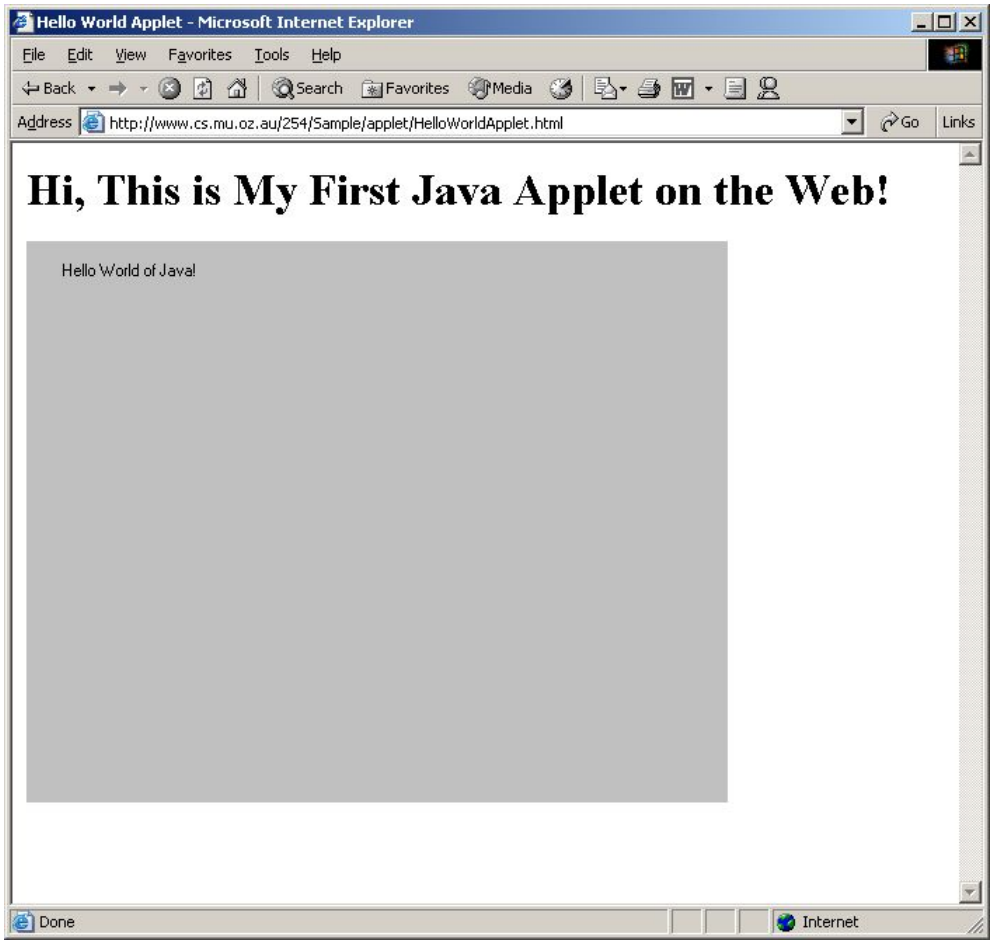
Building Applet Code: An Example

```
//HelloWorldApplet.java
import java.applet.Applet;
import java.awt.*;

public class HelloWorldApplet extends Applet {
        public void paint(Graphics g) {
                g.drawString ("Hello World of Java!",25, 25);
        }
}
```

Embedding Applet in Web Page

```
<HTML>

<HEAD>
<TITLE>
  Hello World Applet
</TITLE>
</HEAD>

<body>
<h1>Hi, This is My First Java Applet on the Web!</h1>
<APPLET CODE="HelloWorldApplet.class" width=500 height=400>
</APPLET>
</body>

</HTML>
```

Accessing Web page (runs Applet)



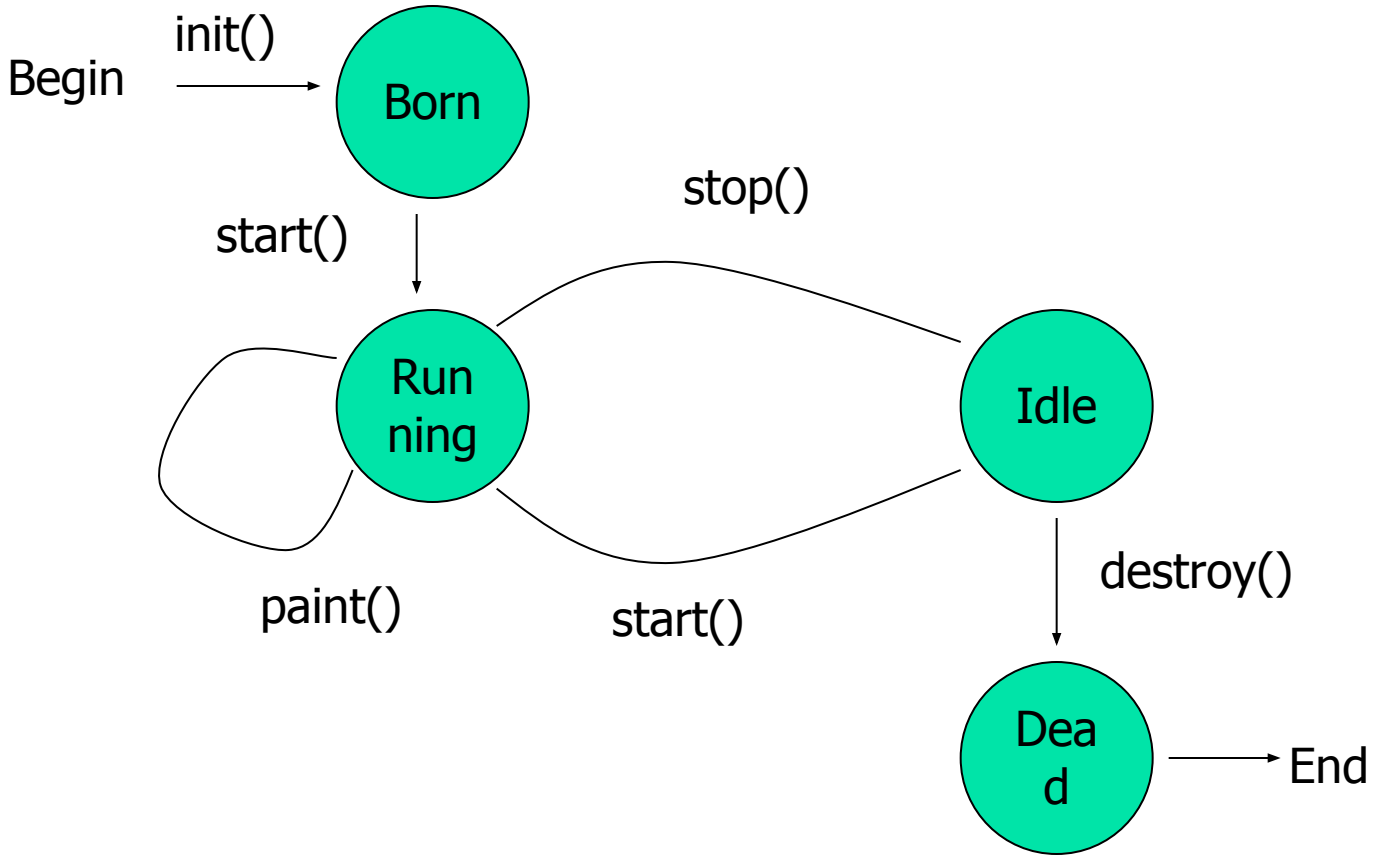
Applet Life Cycle

- Every applet inherits a set of default behaviours from the Applet class. As a result, when an applet is loaded, it undergoes a series of changes in its state. The applet states include:
  - Initialisation – invokes init()
  - Running – invokes start()
  - Display – invokes paint()
  - Idle – invokes stop()
  - Dead/Destroyed State – invokes destroy()

Applet States

- Initialisation – invokes init() – only once
  - Invoked when applet is first loaded.
- Running – invokes start() – more than once
  - For the first time, it is called automatically by the system after init() method execution.
  - It is also invoked when applet moves from idle/stop() state to active state. For example, when we return back to the Web page after temporary visiting other pages.
- Display – invokes paint() - more than once
  - It happens immediately after the applet enters into the running state. It is responsible for displaying output.
- Idle – invokes stop() - more than once
  - It is invoked when the applet is stopped from running. For example, it occurs when we leave a web page.
- Dead/Destroyed State – invokes destroy() - only once
  - This occurs automatically by invoking destroy() method when we quite the browser.

Applet Life Cycle Diagram

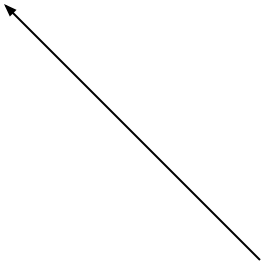


Passing Parameters to Applet

```
<HTML>

<HEAD>
<TITLE>
  Hello World Applet
</TITLE>
</HEAD>

<body>
<h1>Hi, This is My First Communicating Applet on the Web!</h1>
<APPLET
   CODE="HelloAppletMsg.class" width=500 height=400>
   <PARAM NAME="Greetings" VALUE="Hello Friend, How are you?">
</APPLET>

</body>

</HTML>
```

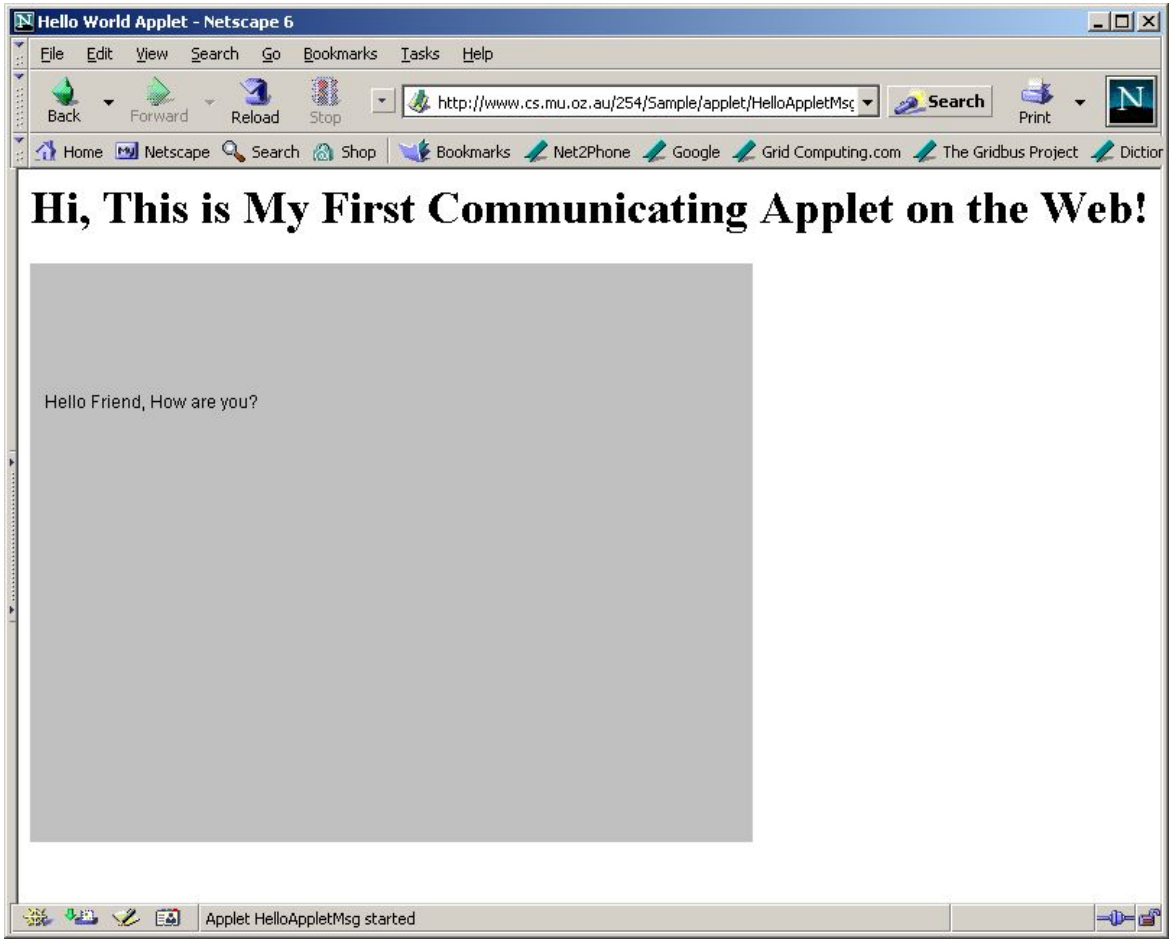
Applet Program Accepting Parameters

```
//HelloAppletMsg.java
import java.applet.Applet;
import java.awt.*;

public class HelloAppletMsg extends Applet {

    String msg;

    public void init()
    {
      msg = getParameter("Greetings");
      if( msg == null)
            msg = "Hello";
    }
    public void paint(Graphics g) {
      g.drawString (msg,10, 100);
    }
}
```



This is name of parameter specified in PARAM tag;
This method returns the value of paramter.

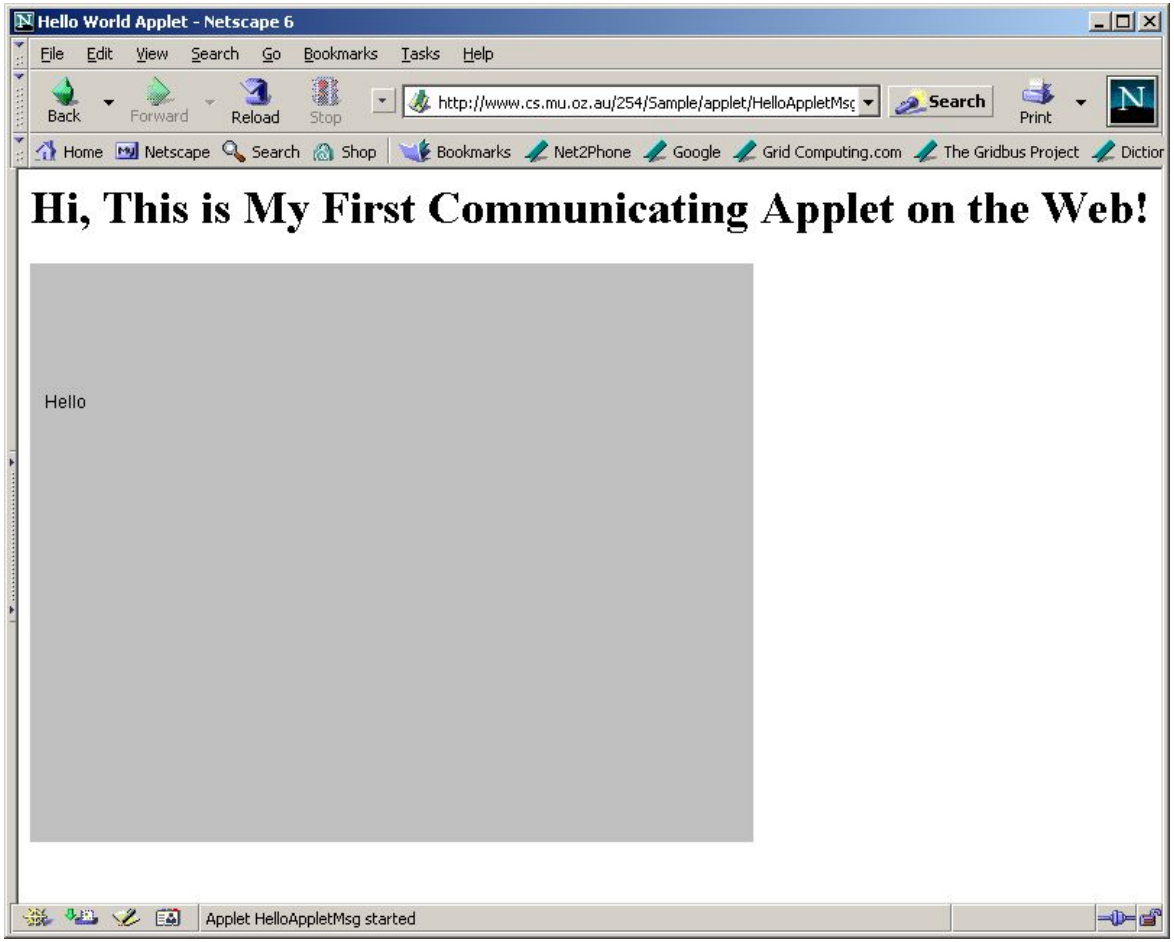
HelloAppletMsg.html



What happen if we don't pass parameter? See HelloAppletMsg1.html

```
<HTML>

<HEAD>
<TITLE>
  Hello World Applet
</TITLE>
</HEAD>

<body>
<h1>Hi, This is My First Communicating Applet on the Web!</h1>
<APPLET
   CODE="HelloAppletMsg.class" width=500 height=400>
</APPLET>

</body>

</HTML>
```

getParameter() returns null. Some default value may be used.



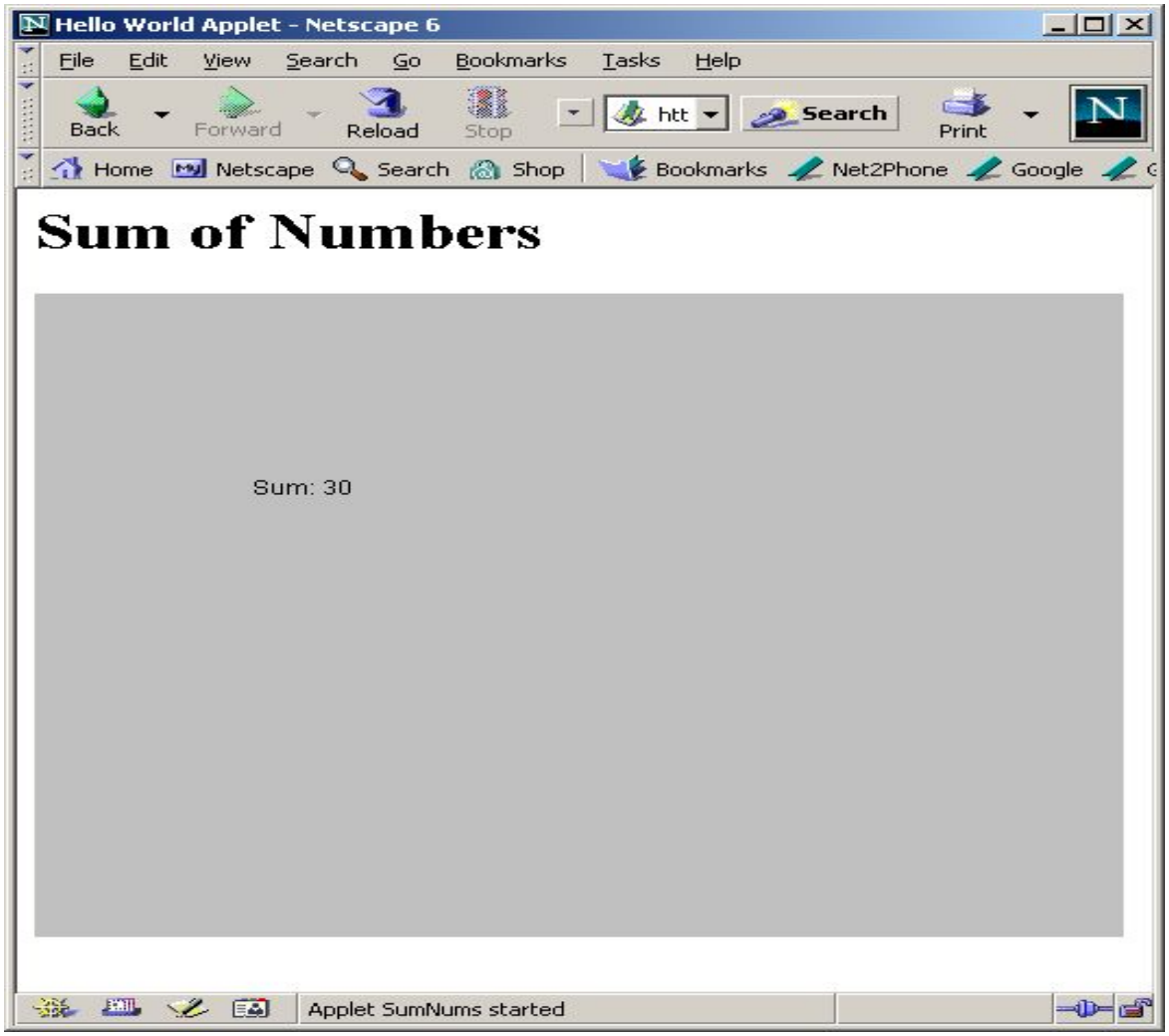
Displaying Numeric Values

```
//SumNums.java
import java.applet.Applet;
import java.awt.*;

public class SumNums extends Applet {
    public void paint(Graphics g) {
      int num1 = 10;
      int num2 = 20;
      int sum = num1 + num2;

      String str = "Sum: "+String.valueOf(sum);
      g.drawString (str,100, 125);
    }
}
```

SunNums.html

```
<HTML>

<HEAD>
<TITLE>
  Hello World Applet
</TITLE>
</HEAD>

<body>
<h1>Sum of Numbers</h1>
<APPLET CODE="SumNums.class" width=500 height=400>
</APPLET>
</body>

</HTML>
```

Applet – Sum Numbers



Interactive Applets

- Applets work in a graphical environment. Therefore, applets treats inputs as text strings.
- We need to create an area on the screen in which use can type and edit input items.
- We can do this using TextField class of the applet package.
- When data is entered, an event is generated. This can be used to refresh the applet output based on input values.

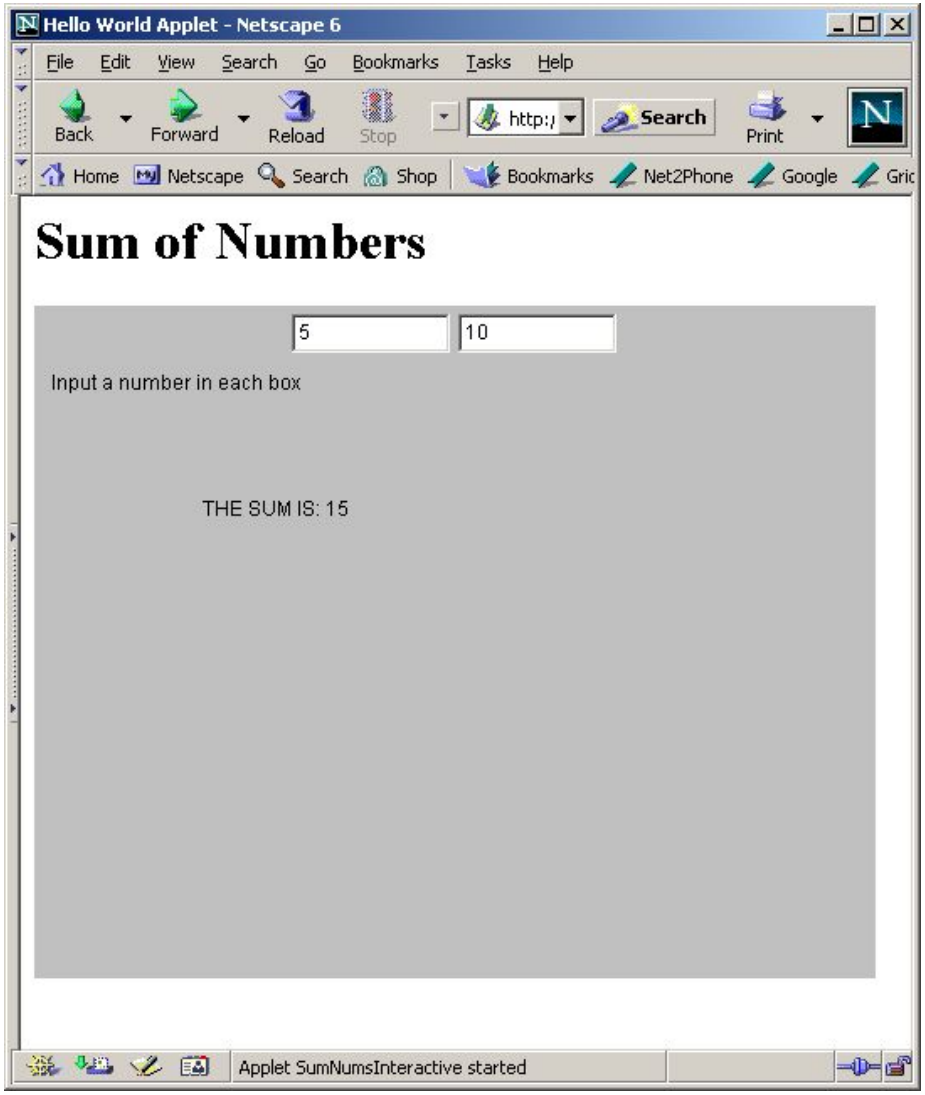
Interactive Applet Program..(cont)

```
//SumNumsInteractive..java
import java.applet.Applet;
import java.awt.*;

public class SumNumsInteractive extends Applet {
        TextField text1, text2;
        public void init()
        {
          text1 = new TextField(10);
          text2 = new TextField(10);
          text1.setText("0");
          text2.setText("0");
          add(text1);
          add(text2);
        }
        public void paint(Graphics g) {
          int num1 = 0;
          int num2 = 0;
          int sum;
          String s1, s2, s3;

          g.drawString("Input a number in each box ", 10, 50);
          try {
                    s1 = text1.getText();
                    num1 = Integer.parseInt(s1);
                    s2 = text2.getText();
                    num2 = Integer.parseInt(s2);
          }
          catch(Exception e1)
          {}
```

Interactive Applet Program.

```
    sum = num1 + num2;
    String str = "THE SUM IS: "+String.valueOf(sum);
    g.drawString (str,100, 125);
   }
   public boolean action(Event ev, Object obj)
   {
    repaint();
    return true;
   }
}
```

Interactive Applet Execution



Summary

- Applets are designed to operate in Internet and Web environment. They enable the delivery of applications via the Web.
- This is demonstrate by things that we learned in this lecture such as:
  - How do applets differ from applications?
  - Life cycles of applets
  - How to design applets?
  - How to execute applets?
  - How to provide interactive inputs?