Java Applet

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.

### Advantage of Applet

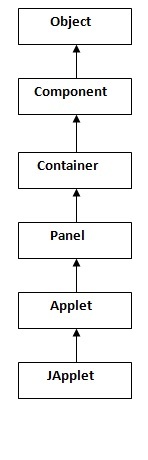
There are many advantages of applet. They are as follows:

* It works at client side so less response time.
* Secured
* It can be executed by browsers running under many platforms, 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. |

### Lifecycle of Java Applet

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

### Lifecycle methods for Applet:

The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides 1 life cycle methods for an applet.

### java.applet.Applet class

For creating any applet java.applet.Applet class must be inherited. It provides 4 life cycle methods of applet.

1. **public void init():** is used to initialized the Applet. It is invoked only once.
2. **public void start():** is invoked after the init() method or browser is maximized. It is used to start the Applet.
3. **public void stop():** is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
4. **public void destroy():** is used to destroy the Applet. It is invoked only once.

### java.awt.Component class

The Component class provides 1 life cycle method of applet.

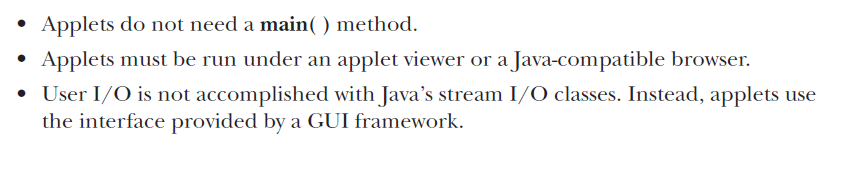
1. **public void paint(Graphics g):** is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

### How to run an Applet?

There are two ways to run an applet

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

### 



### Simple example of Applet by html file:

To execute the applet by html file, create an applet and compile it. After that create an html file and place the applet code in html file. Now click the html file.

1. //First.java
2. **import** java.applet.Applet;
3. **import** java.awt.Graphics;
4. **public** **class** First **extends** Applet{
6. **public** **void** paint(Graphics g){
7. g.drawString("welcome",150,150);
8. }
10. }

### myapplet.html

1. <html>
2. <body>
3. <applet code="First.class" width="300" height="300">
4. </applet>
5. </body>
6. </html>

# Displaying Graphics in Applet

java.awt.Graphics class provides many methods for graphics programming.

## Commonly used methods of Graphics class:

1. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
2. **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
3. **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
4. **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
5. **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.
6. **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).
7. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
8. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
9. **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc.
10. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
11. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

## Example of Graphics in applet:

1. **import** java.applet.Applet;
2. **import** java.awt.\*;
4. **public** **class** GraphicsDemo **extends** Applet{
6. **public** **void** paint(Graphics g){
7. g.setColor(Color.red);
8. g.drawString("Welcome",50, 50);
9. g.drawLine(20,30,20,300);
10. g.drawRect(70,100,30,30);
11. g.fillRect(170,100,30,30);
12. g.drawOval(70,200,30,30);
14. g.setColor(Color.pink);
15. g.fillOval(170,200,30,30);
16. g.drawArc(90,150,30,30,30,270);
17. g.fillArc(270,150,30,30,0,180);
19. }
20. }

### myapplet.html

1. <html>
2. <body>
3. <applet code="GraphicsDemo.class" width="300" height="300">
4. </applet>
5. </body>
6. </html>

**Applet Lifecycle**

package applet\_progs;

import java.awt.Graphics;

import javax.swing.JApplet;

import java.applet.Applet;

public class lc extends Applet {

.

public void init()

{

showStatus("The applet is initializing...");

for(int i=0;i<100000000;i++);

}

public void start()

{

showStatus("The applet is starting...");

for(int i=0;i<100000000;i++);

}

public void stop()

{

showStatus("The applet is stopping...");

for(int i=0;i<100000000;i++);

}

public void destroy()

{

showStatus("The applet is destroying...");

for(int i=0;i<100000000;i++);

}

}

**Parsing Parameter to Applet**

package applet\_progs;

import java.applet.Applet;

import java.awt.Graphics;

public class a4 extends Applet {

String message;

int firstNumber;

double secondNumber;

public void init() {

message = getParameter("displayMessage");

String num1 = getParameter("number1");

String num2 = getParameter("number2");

firstNumber = Integer.parseInt(num1);

secondNumber = Double.parseDouble(num2);

}

public void paint(Graphics g) {

g.drawString(message, 50, 60);

g.drawString("Flexible code: Product of two numbers: " + firstNumber \* secondNumber, 50, 80);

}

}

**HTML File**

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

Java Applet

<applet code="DemoApplet.class" width="300" height="200">

<param name="displayMessage" value="Hello World">

<param name="number1" value="10">

<param name="number2" value="20.5">

</applet>

</body>

</html>

**P10\_1**

import java.applet.Applet;

import java.awt.Graphics;

public class a2 extends Applet {

public void paint(Graphics g) {

g.drawString("Hello World", 50, 75);

showStatus("Your Applet is running..!! Greetings");

}

}

**P10\_2**

import java.applet.Applet;

import java.awt.Color;

import java.awt.Graphics;

public class a1 extends Applet{

public void paint(Graphics g)

{

g.setColor(Color.blue);

g.fillRect(20, 20, 80, 80);

g.setColor(Color.red);

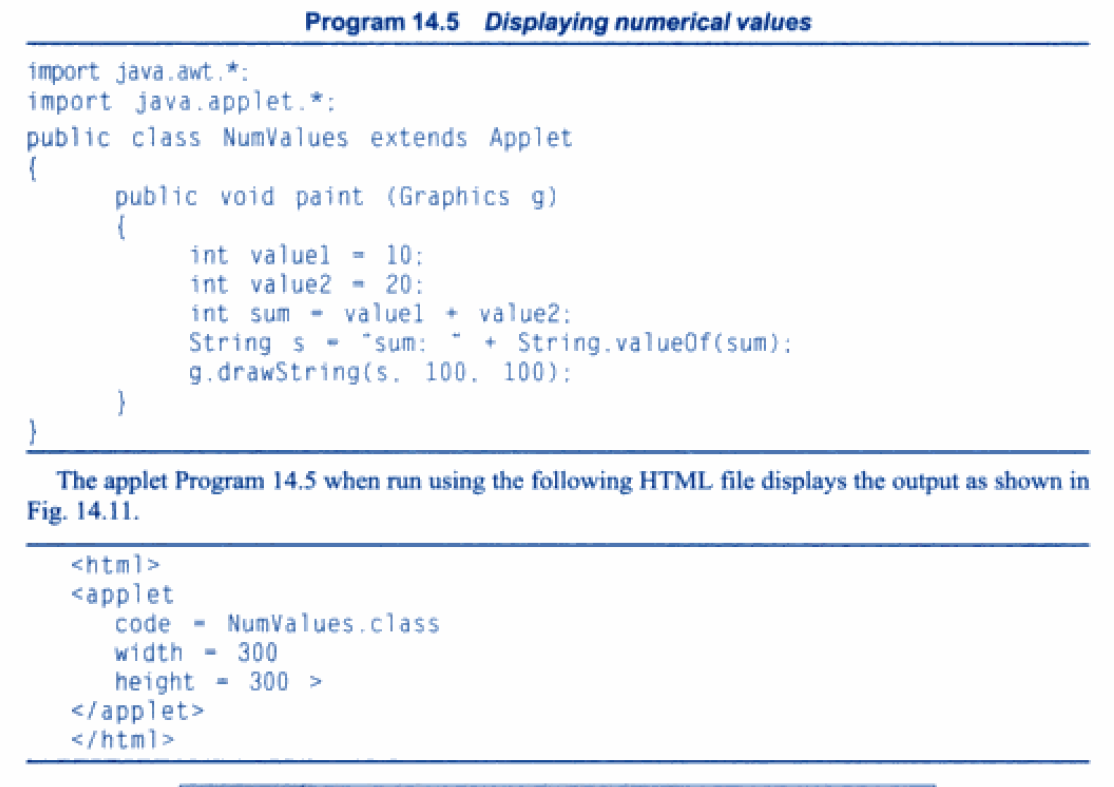
g.fillRect(35, 35, 50, 50);

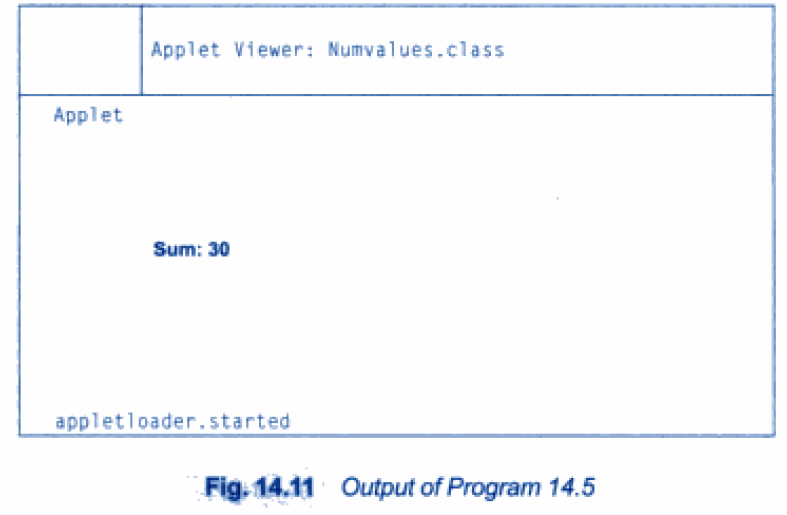
g.setColor(Color.green);

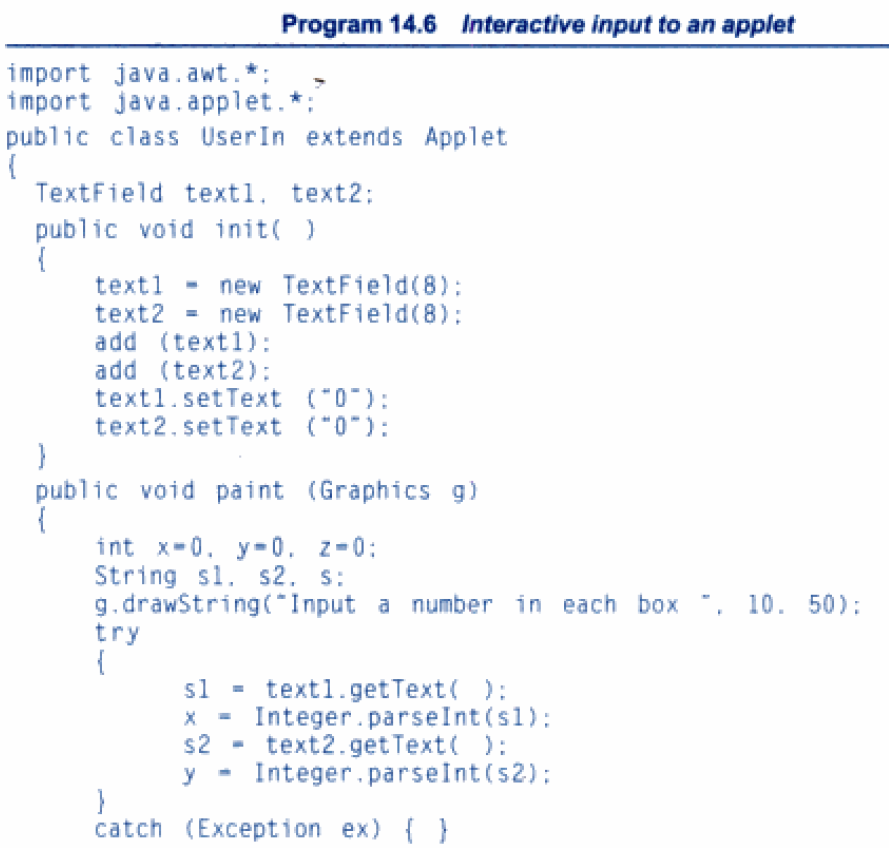
g.fillRect(45, 45, 25, 25);

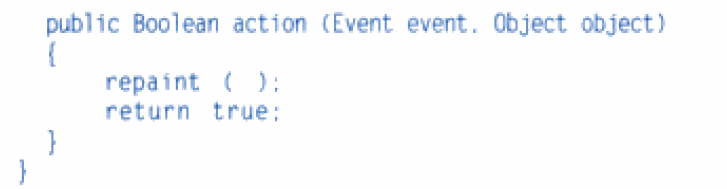
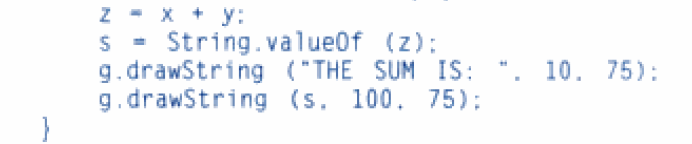
}

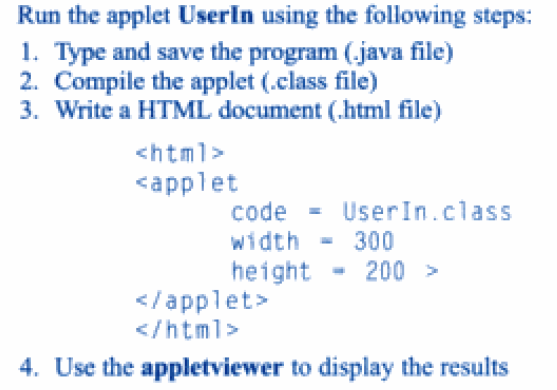
}

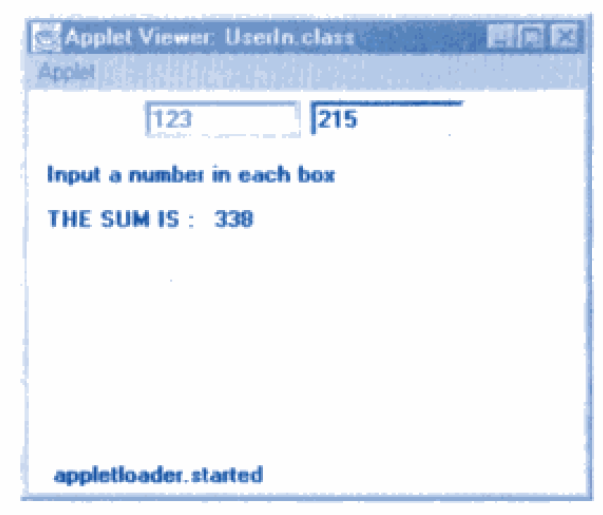












# EventHandling in Applet

|  |
| --- |
| As we perform event handling in AWT or Swing, we can perform it in applet also. Let's see the simple example of event handling in applet that prints a message by click on the button. |

## Example of EventHandling in applet:

1. **import** java.applet.\*;
2. **import** java.awt.\*;
3. **import** java.awt.event.\*;
4. **public** **class** EventApplet **extends** Applet **implements** ActionListener{
5. Button b;
6. TextField tf;
8. **public** **void** init(){
9. tf=**new** TextField();
10. tf.setBounds(30,40,150,20);
12. b=**new** Button("Click");
13. b.setBounds(80,150,60,50);
15. add(b);add(tf);
16. b.addActionListener(**this**);
18. setLayout(**null**);
19. }
21. **public** **void** actionPerformed(ActionEvent e){
22. tf.setText("Welcome");
23. }
24. }

|  |
| --- |
| In the above example, we have created all the controls in init() method because it is invoked only once. |

### myapplet.html

1. <html>
2. <body>
3. <applet code="EventApplet.class" width="300" height="300">
4. </applet>
5. </body>
6. </html>