

(Java Programming -22412)

Unit V : Java Applets and Graphics Programming.

(Weightage-10marks)

Pranjalsare(SYCA)

Applet

Applets are small programs that are primarily used in Internet computing.

They can be transported over to the internet from one computer to another computer and run ~~the using~~ Applet viewer or any web browser supported by Java.

It can perform arithmetic operations, display graphics, play sounds, accept user input, create animation.

Java has enabled interactive multimedia web documents. A web page now contains only a simple text or static image but also a Java applet which can produce graphics, sound and moving images.

Java applets therefore have begun to make a significant impact on www.

Normal Program → Application

Why we need Applet?

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

Java

Application programming

Applet programming

Java Application

Application use main() method for execution.

Application can run independently.

Application are control driven.

Application can read from or write to files in computer.

Application can run any program from Local computer.

Applications are not restricted of using other language libraries.

Java Applet

Applet does not use main() method for execution.

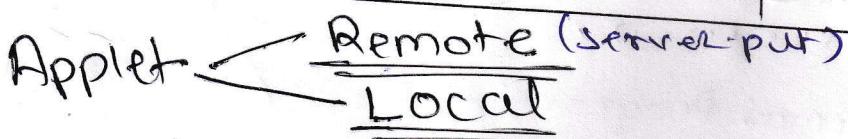
Applets can't run independently.

Applets are eventdriven

Applets cannot read or write to files in Local Computer.

Applet can't run any program from Local computer.

Applets restricted from using libraries like C and C++.



An applet developed locally and stored in a local system is known as local applet. It does not require internet as it works in same system. When you run local applet it loads and locates the directories of specified applets.

A remote applet is that which is developed by someone else and stored on a remote computer connected to the Internet. If we are connected to internet, we can download the remote applet onto our system via the Internet and run it. In order to know the applet, we should know remote address (URL).

Writing JAVA Applet Code

- ① Building an applet code (.java file)
- ② Creating executable applet (.class file)
- ③ Design applet ~~using~~ / web page using
HTML tags.
- ④ Prepare Applet tag. (<Applet>)
- ⑤ Incorporating <Applet> tag into web page.
- ⑥ Create HTML file.
- ⑦ Testing applet code.

Building an applet Code (.java file)

In order to create an applet tag, we have to use two classes ~~packages~~ Applet & Graphics from Java class library.

The Java Applet class contains in `java.applet` package provides life and behaviour to applet through its methods such as `init()`, `start()` & `paint()`.

Unlike application in which execution starts from `main()` function. In Applet, Java automatically calls a series of Applet class methods for running and stopping the applet code when Applet is loaded. The Applet class mains the life cycle.

The `paint()` method of applet class, when it called displays result of applet code on screen.

The output may be text, graphics or sound - the paint method requires a `Graphics` object as an argument.

```
public void paint ( Graphics g )  
this code imports java.awt contained in the  
Graphics class.
```

Structure

```

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

public class HelloJava extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hello JAVA", 10, 100);
    }
}

```

Creating an executable applet (-class file)

Now after typing the code, you have to save the file with any name with extension .java and right after that you have to convert the .java file to executable file i.e. class file.

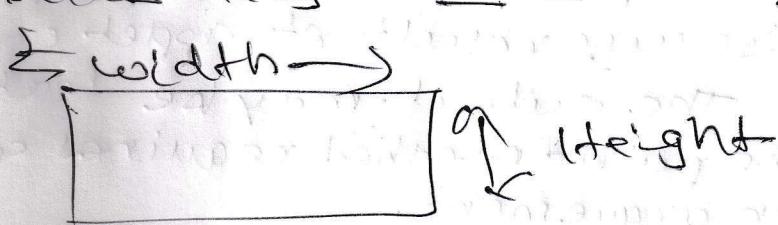
Designing web page using HTML tags

In order to run Java Applet, we need to create an HTML file and associate applet code with HTML file.

Prepare <applet> tag

To incorporate Applet code to HTML document we use <applet> tag.

Applet code = filename.class height= width=

<applet> 

Incorporating <applet> tag into webpage

```

<HTML><head>
<TITLE> Hello welcome </TITLE>
</head>
<Body>
<applet code="Hello.class" height=400 width=400>
</applet></body></HTML>.

```

Running the Applet

In order to run Applet we must have all these files three under same directory.

hellojavaa.java

hellojavae.class

hellojavae.html

Applet viewer

we can run in two ways

web browser

If we use a Java enabled web browser we will be able to see the entire webpage containing the applet. If we use tool of appletviewer we will only see the applet output.

Applet life Cycle

Every Java applet has a life cycle through which it passes through out his life span. Every java applet inherits the behaviour from Applet class.

It contains following steps/states.

- ① Born or Initialization state
- ② Running state
- ③ Idle state
- ④ Dead or Destroyed state.

init()

Begin



start()

Display



paint()

stop()

start()

Idle

destroy()

Dead

Initialisation State

When applet is loaded, it enters the initialization state. This happens by calling the `init()` method of Applet class. Here, the Applet is born. At this stage the following action may be taken:

- ① Create object as required by Applet.
- ② Set up initial value.
- ③ Load images or fonts.
- ④ Set up colours.

Later initialization occurs only once in applet's life cycle. To go of any above point we need to override `init()` method.

```
public void init()
```

E

Initialisation

→

Y

Running State

Applet enters the running state when the system calls the `start()` method of Applet class. This occurs automatically after the applet is initialized.

Suppose we leave the web page containing the applet temporarily to another page and return back to page. Then again start applet running. Note that, unlike `init()` method, the `start()` method can call more than once.

```
public void start()
```

S

Start

→

Stop

Idle or stopped state

An Applet becomes idle when it is stopped from running. Stopping occurs automatically when we leave current web page on Applet which running by calling `stop()`.

public void stop()

{

//Code

}

Dead State

An Applet is said to be dead when it is removed from memory. This occurs automatically by invoking the destroy() method when we quit browser.

Like initialization, destroying occurs only once in applet Life Cycle.

public void destroy()

{

 //Code

}

For Display State

Whenever an Applet has to perform some output operation, it enters Display State. This happens after an Applet enters to running state.

The paint() method is called to accomplish this task.

public void paint(Graphics g)

{

 //Action

}

Note

"Display State" is not considered as the part of life cycle in Applet Life Cycle.

paramtag <param>

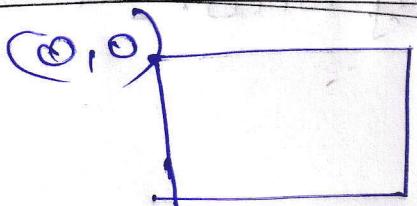
Syntax:

```
<param name="name" value="value">
```

Example:

```
<param name="color" value="red">  
<param name="xvalue" value="20">
```

Graphics programming (java.awt)



Abstract
Windows
Toolkit

→ drawString (String str, int x, int y)
Display + edit

→ drawLine (int x1, int y1, int x2, int y2)
Display Line

→ drawRect (int x, int y, int width, int height)
width = height = square

→ drawOval (int x, int y, int width, int height)
width = height = circle

→ drawArc (int x, int y, int height, int width,
int startAngle, int ArcAngle)

→ drawPolygon (x[], y[], no. of sides)

For colour Fill instead of Draw

Draw Fill Arc()

program to Add two number using Applet

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

public class ABC extends Applet implements
ActionListener.

{

```
Textfield t1 = new Textfield("0");
```

```
Textfield t2 = new Textfield("0");
```

```
Textfield t3 = new Textfield("0");
```

```
Label l1 = new Label("First num");
```

```
Label l2 = new Label("Second num");
```

```
Label l3 = new Label("Sum :");
```

```
Button b = new Button("ADD");
```

public void init()

{

```
add(t1)
```

```
add(l1)
```

```
add(t2)
```

```
add(l2)
```

```
add(t3)
```

```
add(l3)
```

```
add(b)
```

```
b.addActionListener(this);
```

↳ public void actionPerformed(ActionEvent e)

↳ if(e.getSource() == b)

{

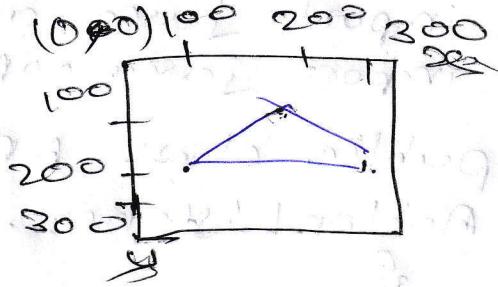
```
int n1 = Integer.parseInt(t1.getText());
```

```
int n2 = Integer.parseInt(t2.getText());
```

```
t3.setText(" " + (n1+n2));
```

23 3
<applet code=ABC.class width=100 height=100>

write a program to generate following output
using drawLine() method.



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

class Triangle extends Applet

{

public void paint(Graphics g)

{
g.setColor(Color.Red);

g.drawLine(100, 200, 200, 100)

g.drawLine(200, 100, 300, 200)

g.drawLine(300, 200, 100, 200)

z
y

Applet code = "Triangle.class" height=400
width=400>

(applet)

Write a program to draw a chess board in
Applet.

→ ~~import java.awt.*;~~
~~import java.applet.*;~~

~~public class Chess extends Applet~~

~~static int n = 10;~~

~~public void paint(Graphics g)~~

~~int i, j;~~

```
import java.applet.*;
import java.awt.*;
public class extChess extends Applet
{
    private final int rows = 8;
    private final int cols = 8;
    private final int tileSize = 50;
    public void paint(Graphics g)
    {
        for (int row = 0; row < rows; row++)
        {
            for (int col = 0; col < cols; col++)
            {
                int x = col * tileSize;
                int y = row * tileSize;
                if ((row + col) % 2 == 0)
                {
                    g.setColor(Color.white);
                }
                else
                {
                    g.setColor(Color.black);
                }
                g.fillRect(x, y, tileSize, tileSize);
            }
        }
    }
}
```