

Applet

- Applet is a special type of program that is embedded in the webpage, to generate dynamic content (HTML program)
- It runs in a java compatible web browser (where jvm is there)
- It works at client machine.

App

- It is a java class which runs in a java compatible web browser.
 - Every Applet class is a sub class of Applet class.
- Difference between Applet and Java standalone app

1. An applet is a java class that extends the `java.applet.Applet` class.

Exa

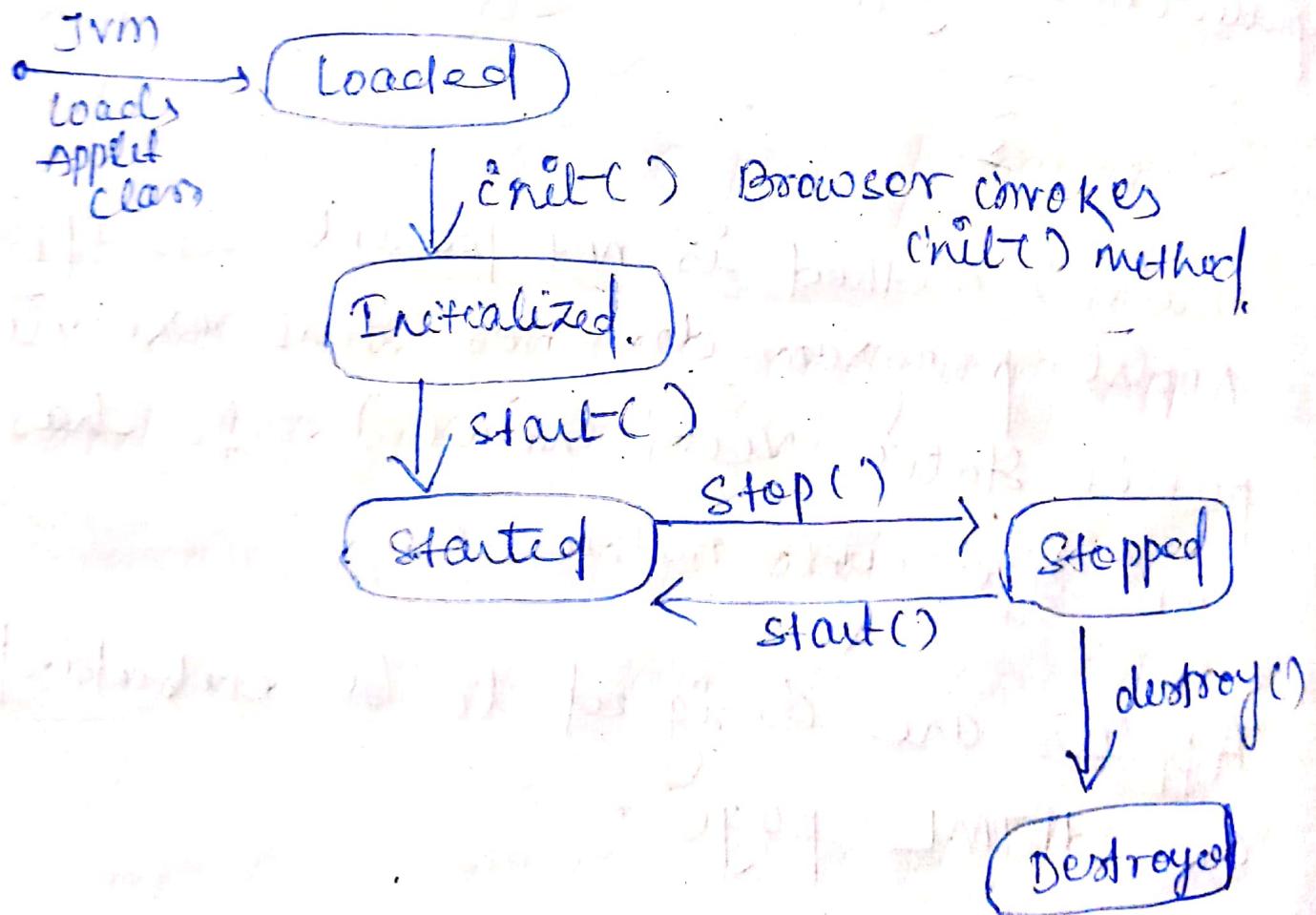
```
public class myApplet extends Applet  
{  
    }  
    
```

2. main() method is not present in applet program.
Applet program does not start execution from public static void main(). It has its own life cycle methods.
3. Applets are designed to be embedded within an HTML page.
4. When a user views an HTML page that contains an <applet> code, then that is downloaded to the user's machine.

5. Then JVM is required to execute an applet either by a plug-in software or by separate runtime environment.
6. The JVM on the user's machine creates an instance or object of the applet class and invokes various lifetime methods of applet.
7. Applets are secure program i.e it has strict security rules enforced by the web browser.

Life Cycle of Applet with Life cycle methods

State chart diagram



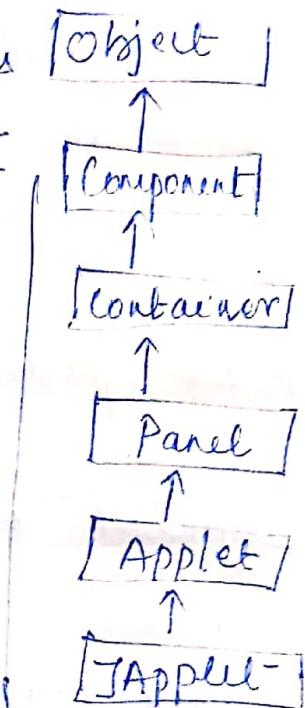
For creating any Applet java.applet.Applet class must be inherited.

It provides 4 life cycle methods of applet.

Hierarchy of Applet class

Inheritance

java.awt.Component class provides life cycle methods which are ultimately methods of Applet class by the virtue of inheritance.



b) Lifecycle

1. Applet is initialized ie

public void init() method is

invoked to initialize the applet

It is called only once.

2. Applet is started ie

public void start()

It is invoked after the init() method or

browser is maximized.

It is called to start the applet.

3. Applet is stopped ie

public void stop() - It is called to

stop the applet. It is called when Applet is stop or browser is minimized.

4. Applet is destroyed ie

public void destroy() - used to destroy the applet object. It is called once.

public void paint(Graphics g)

— It is important method which is the method of Component class.

It is used to render or display any text, geometrical objects like circle, oval, line, rectangle on the applet.

— It provides Graphics class which has the methods to draw any symbol on the applet.

paint() method is invoked immediately after start().

after start() method is called and also anytime the applet needs to be repaint itself in the browser.

How to write an Applet program.

(i) create a java file containing Applet code and life cycle methods.

(ii) create a HTML file and embed the class file of java class created in first step.

(iii) Run applet using either of following

(a) open the HTML file in java compatible web browser.

(b) By using Appletviewer tool.

Example (class must be declared public)

(I) // To display welcome message .

MyApplet.java

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

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

```
public class myApplet extends Applet
```

```
{ public void paint(Graphics g)
```

```
{
```

```
    g.drawString("Welcome to my Applet", 100,
```

```
    100);
```

```
}
```

```
}
```

→ Save the program in myApplet.java.

(II) Code for HTML file .

→ javac MyApplet.java

(II) Code for HTML file .

→ myApplet.html

```
<html>
```

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

```
</applet>
```

```
</html>
```

(III) Run the HTML file by

opening the myApplet.html in a browser .

Another way of writing Applet program (without using separate html file)

→ open Notepad and write the code -

```
import java.applet.*;  
import java.awt.*;  
  
public class MyApplet extends Applet  
{ public void paint(Graphics g)  
{ g.drawString("Welcome", 100, 100);  
 }  
 }  
  
/*  
<applet code="MyApplet.class" width=500  
 height=500>  
</applet>  
*/
```

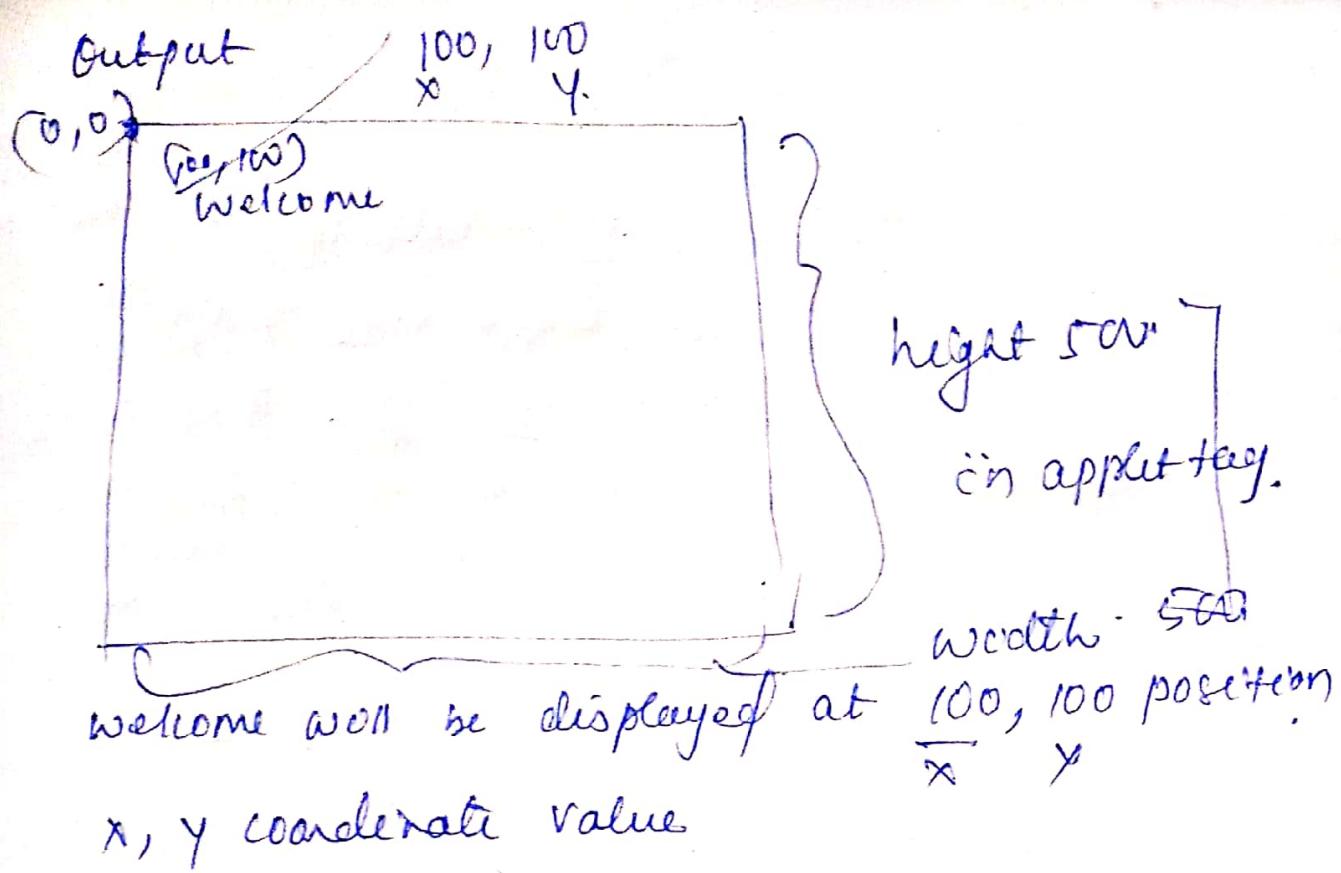
→ save in myApplet.java

→ compile

```
javac MyApplet.java
```

→ run by appletviewer.

```
> appletviewer myApplet.java
```



g.drawString("String to be displayed", x, y).

~~String~~

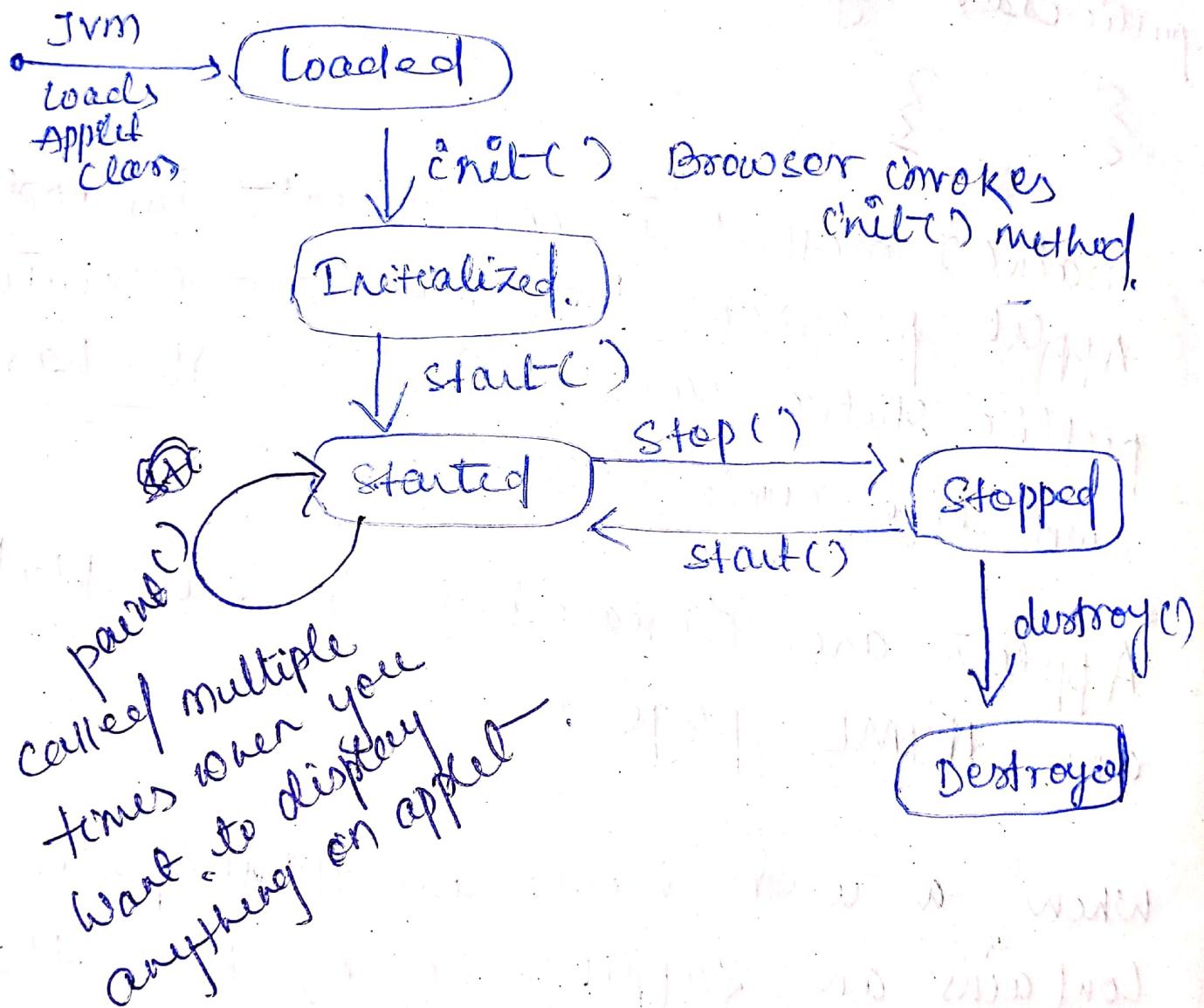
→ drawstring() method. is used to display a string on the applet at x, y position whose initial value is (0,0) starting point of browser.

→ It is a method of Graphics class.

Graphics class present in awt package.

So import java.awt.*; is required

State chart diagram



Methods of Graphics Class

In GUI application, Graphics class can be used to draw different graphics like line, rectangle, circles, polygons, images etc.

How to draw a line:

1. public void drawLine(int x1, int y1, int x2, int y2)



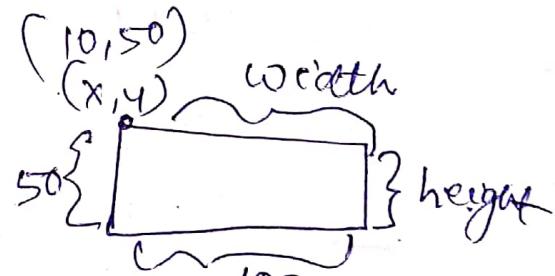
public void paint(Graphics g)

{

g.drawLine(10, 50, 100, 50);

}

2. To draw Rectangle



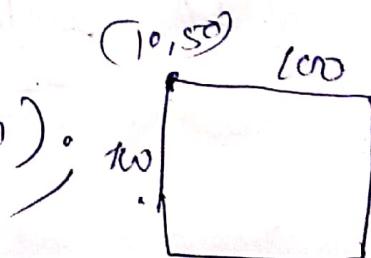
public void drawRect(int x, int y, int w, int h)

g.drawRect(10, 50, 100, 50);

3.

To draw Square

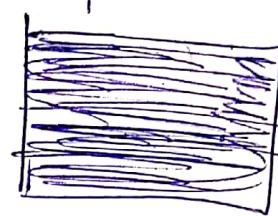
g.drawRect(10, 50, 100, 100);



3. To draw filled Rectangle

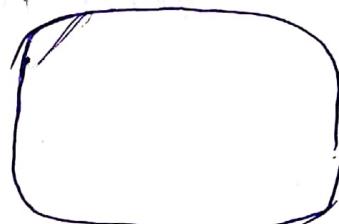
void fillRect(int x, int y, int w, int h)

g.fillRect(10, 50, 100, 100);



4. To draw Rounded Rectangle

void drawRoundRect(int x,
int y, int w, int h, int xdem, int ydem);



void fillRoundRect(int x, int y, int w, int h,
int xdem, int ydem);

5. To draw Oval/Circle



void drawOval(int x, int y, int w, int h)

→ g.drawOval(10, 10, 100, 50);

To draw Circle (both width and height same)

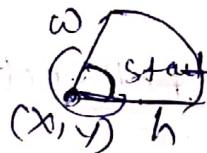
→ g.drawOval(10, 10, 50, 50);

6. Filled Oval

g.fillOval(10, 10, 100, 50);



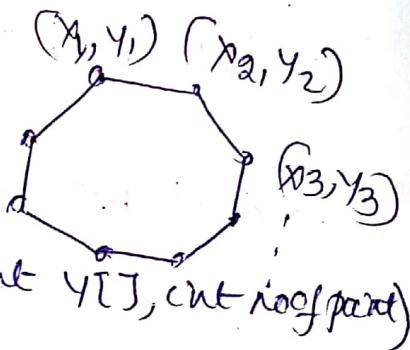
To draw Arc



void drawArc(int x, int y, int w,
int h, int startangle, int sweepangle)

8. void fillArc(int x, int y, int w, int h,
int start, int sweep)

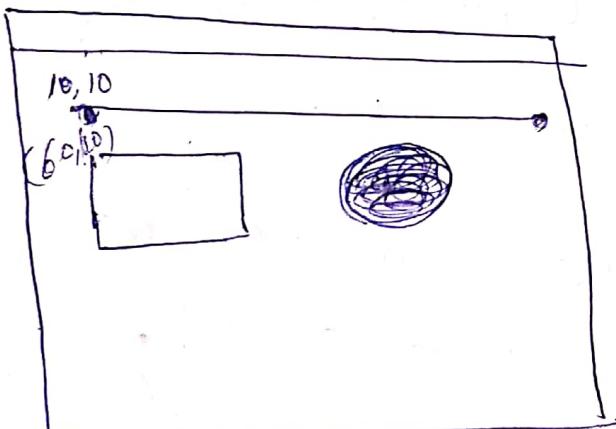
To draw Polygon



void drawPolygon(int x[], int y[], int nof point)

void fillPolygon(int x[], int y[], int nof point)

Exa Write an applet program to



draw a line,
rectangle and circle.

```
import java.awt.*;
import java.applet.Applet;
public class DrawGraphers extends Applet
{ }
```

```
public void paint(Graphics g)
{
    g.drawLine(10, 10, 710, 10);
    // x axis's value increase y axis constant.
    g.drawRect(10, 60, 200, 100);
    // g.drawoval(650, 60, 100, 100);
    g.filloval(650, 60, 100, 100);
}
```

```
/*
<applet code="DrawGraphics.class"
        width=1000, height=600>
```

```
</applet>
*/
```

Save in file DrawGraphics.java.

→ Compile javac DrawGraphics.java

→ Run appletviewer DrawGraphics.java

Note : Coordinate value you have to calculate.

Starting pixel is (0, 0).

End pixel depends on screen resolution (1024, 1024) say

Passing Parameters from Applet Code / HTML code to Java

<param> tag is used to pass parameter value from applet code.

```
/> <applet code="class filename  
width=100 height=1000>  
<param name="X" value="10">  
</applet> </>
```

How to access <param> value in java applet?

There is a method getParameter() in Applet class which is used to access parameter value.

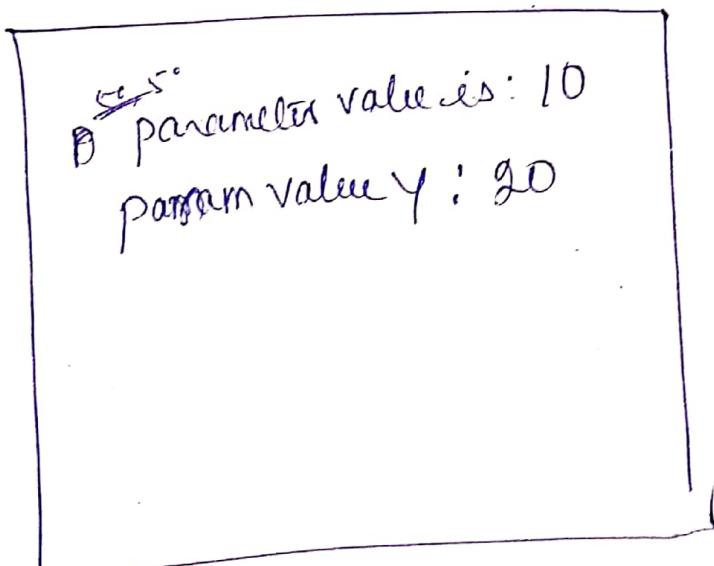
```
String getParameter(String Paramname)
```

Exa

```
import java.awt.*;  
import java.applet.*;  
public class paramDemo extends Applet  
{  
    String s1,s2;  
    public void init()  
    {  
        s1 = getParameter("X");  
        s2 = getParameter("Y");  
    }  
}
```

```
public void paint(Graphics g)
{
    g.drawString("parameter value is: " + s,
                50, 50);
    g.drawString("param value y: " + $1,
                50, 100);
}
/*<applet code="ParamDemo" width=500
height=500>
<param name="x" value=10>
<param name="y" value=20>
</applet>*/
```

O/P



Moving a circle from left to right

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

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

```
public class movecircle extends Applet implements  
Runnable
```

{

```
int x, flag;
```

```
Thread t;
```

```
public void init()
```

{

```
x = 0;
```

// current class object
// is passed

```
t = new Thread(this);
```

```
t.start();
```

}

```
public void run()
```

{

```
while(true) // infinite loop
```

```
{ if (x == 0)
```

```
flag = 0;
```

```
if (x == 450)
```

```
flag = 1;
```

```
if (flag == 0)
```

```
x = x + 5;
```

```
if (flag == 1)
    x = x - 5;
    repaint(); // calls paint() method with
                updated value
try
{
    Thread.sleep(100);
}
catch(InterruptedException e)
{
}
}

// end of while
}

// end of run

public void paint(Graphics g)
{
    g.drawOval(x, 100, 50, 50);
}
```

repaint()

repaint() method is method of `java.awt.Component` class used to repaint the window or applet.
 repaint() → calls update() → calls paint()
 → repaint() method calls paint() method automatically with update of new value passed in drawX() methods.

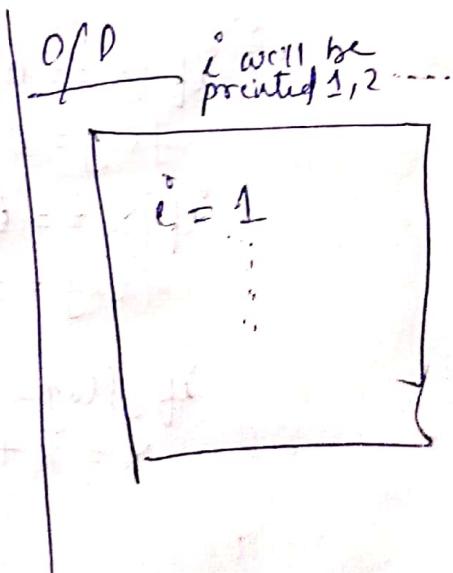
Exa

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

public class repaintDemo extends Applet
{
    int i;
    public void paint(Graphics g)
    {
        g.drawString("i = " + i, 100, 100);
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException e)
        {
            i++;
            repaint();
        }
    }
}
```

```
/*<applet code="repaintDemo"
width=500 height=500>
```

```
</applet> */
```



How to set the color (Background and Foreground (font color)) of an application

Every color value is a static variable in Color class.

Color class present in `java.awt` package.

`Color.BLACK`

`Color.RED`

or

Color can be set through

`rgb(255, 0, 0)`

create color object

`Color c = new Color(Color.
RED);`

`setBackground(c);`

Methods

setBackground(Color value) to change background color

setColor(Color value) to change foreground color

Exa

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

public class ChangeColor extends Applet
{
    public void init()
    {
        setBackground(Color.CYAN);
    }

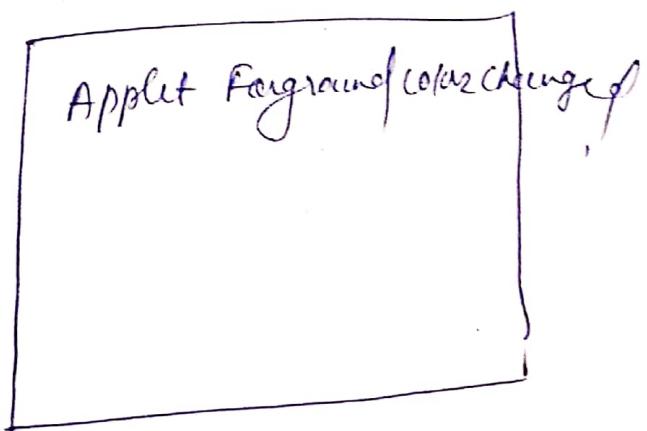
    public void paint(Graphics g)
    {
        g.setColor(Color.RED);
        g.drawString("Applet foreground  
color changed"), 50, 50);
    }
}

<applet code = "ChangeColor.class" width = 520  
height = 500>
```

</applet>

* /
OP *

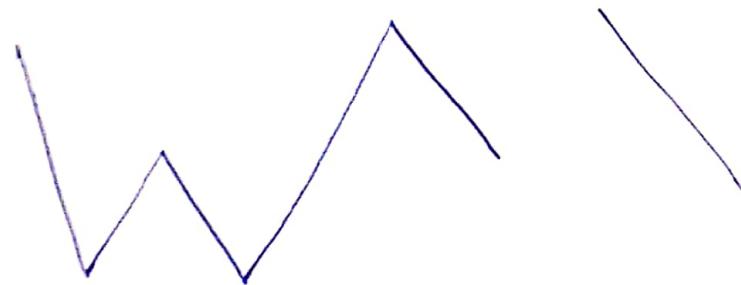
Background will be
in cyan color
and Text will be in
red color



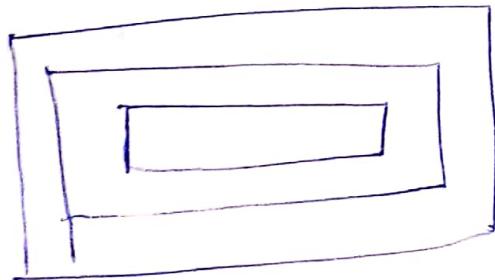
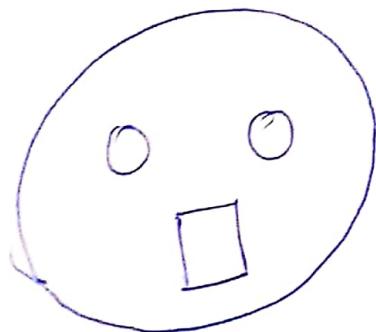
2. create applet to demonstrate life cycle methods



2.



3. background color black, foreground white



04 Move a ball horizontally through screen.

very very