Experiment No._____ **Date___/___/2020**

TITLE OF EXPERIMENT: - A program in Java to learn graphics Class

DIVISION:_____ **BRANCH: _____**

BATCH:_____ **ROLL NO.: _____**

PERFORMED ON DATE: _____

SIGNATURE OF TEACHING STAFF:

EXPERIMENT NO. 11

Aim: Write a Java program to draw oval, rectangle, line, text using graphics class

Objective: To learn graphics class in Java

Software:

1. Eclipse
2. JDK 16

Theory:

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

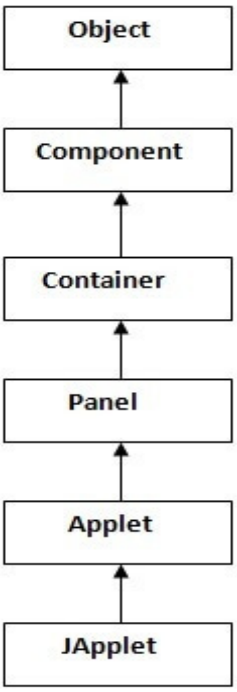
There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

- Plugin is required at client browser to execute applet.

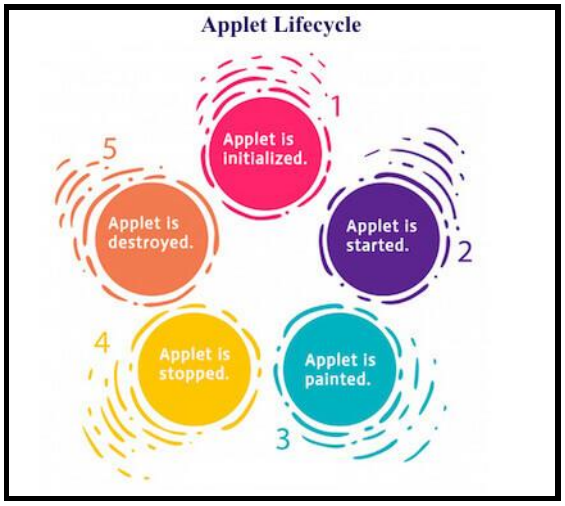
Hierarchy of Applet



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).

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.

Java Swing

Java Swing tutorial is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Difference between AWT and Swing

There are many differences between java awt and swing that are given below.

No.	Java AWT	Java Swing
1)	AWT components are **platform-dependent**.	Java swing components are **platform-independent**.
2)	AWT components are **heavyweight**.	Swing components are **lightweight**.
3)	AWT **doesn't support pluggable look and feel**.	Swing **supports pluggable look and feel**.
4)	AWT provides **less components** than Swing.	Swing provides **more powerful components** such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
5)	AWT **doesn't follows MVC**(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.	Swing **follows MVC**.

JFC:

The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.

Program:

1. Draw a line in Java Applet:

The line is the simplest shape that we can draw with the Graphics class.
The **drawLine()** method takes two pair of coordinates (x1, y1) and (y1, y2) as arguments and draws a line between them. It has the following syntax:

```
g.drawLine(x1, y2, x2, y2); // g is the Graphics object passed to paint() method.
```

drawLine(int x1, int y1, int x2, int y2)

Parameters: The drawLine method takes four arguments:

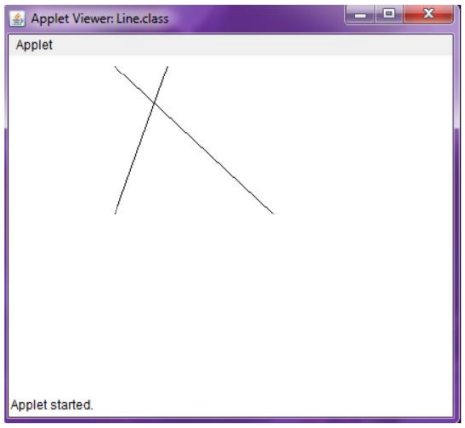
- **x1** – It takes the first point's x coordinate.
- **y1** – It takes first point's y coordinate.
- **x2** – It takes second point's x coordinate.
- **y2** – It takes second point's y coordinate

This method will draw a line starting from (x1, y1) co-ordinates to (x2, y2) co-ordinates.

```
import java.awt.*;
import java.applet.*;
public class Line extends Applet
{
public void paint(Graphics g)
{
g.drawLine(100,10,250, 150);
g.drawLine(100,150,150,10);
}
}
```

```
<html>
<head>
</head>
<body>
<applet code = "Line.class" width = "420" height = "320"></applet>
</body>
</html>
```

Output:

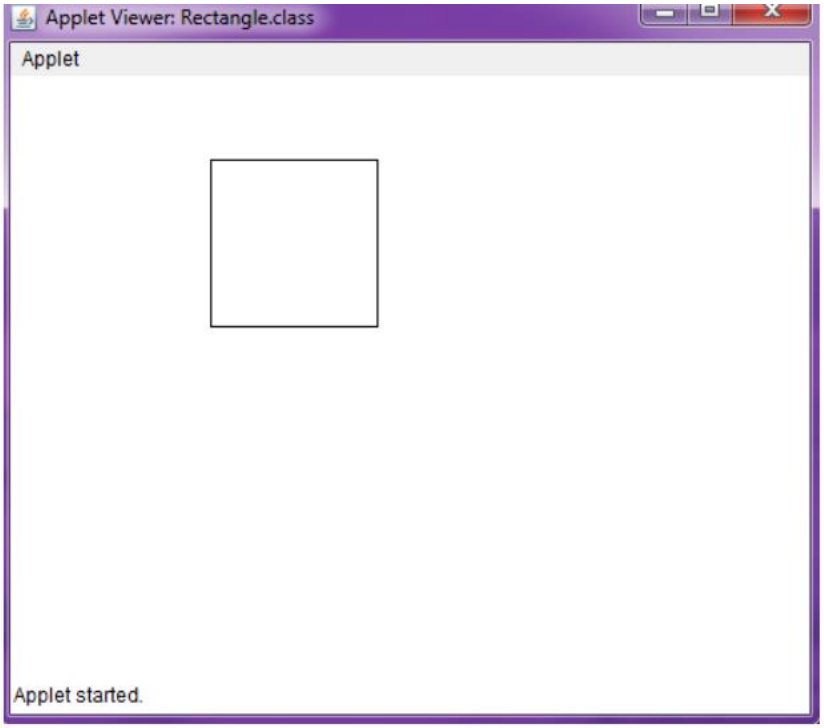


2. Draw a rectangle in Java Applet:

```
import java.awt.*;
import java.applet.*;
public class Rectangle extends Applet
{
public void paint(Graphics g)
{
g.setColor(Color.black);
g.drawRect(120, 50, 100, 100);
}
}
```

```
<applet code = "Rectangle.class" width = "480" height = "360"></applet>
```

Output:



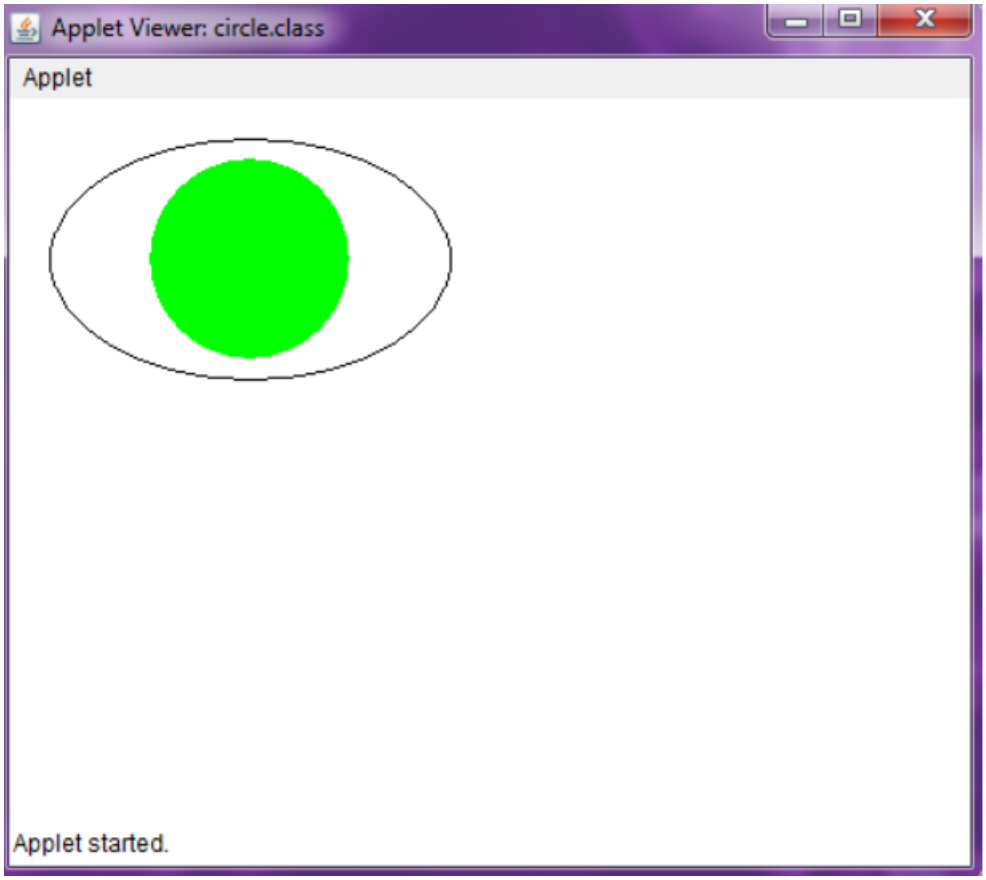
3. Draw Circles and Ellipses in Java Applet

In Java, the Graphics class doesn't have any method for circles or ellipses. However, the **drawOval()** method can be used to draw a circle or an ellipse. Ovals are just like a rectangle with overly rounded corners.
The **drawOval()** method takes four arguments: the first two represent the top-left corner of the imaginary rectangle and the other two represent the width and height of the oval itself.
Note: If the width and height are the same, the oval becomes a circle. The oval's coordinates are the coordinates of an enclosing rectangle.

```
import java.awt.*;
import java.applet.*;
public class circle extends Applet
{
public void paint(Graphics g)
{
g.drawOval(20,20,200,120);
g.setColor(Color.green);
g.fillOval(70,30,100,100);
}
}
```

```
<html>
<head>
</head>
<body>
<applet code = "circle.class" width = "480" height = "360"></applet>
</body>
</html>
```

Output:



Conclusion:

Screenshot's of Program and Result: