JAVA
GRAPHICS PROGRAMMING

Presented by
Pooja Rani
GNKC, YNR

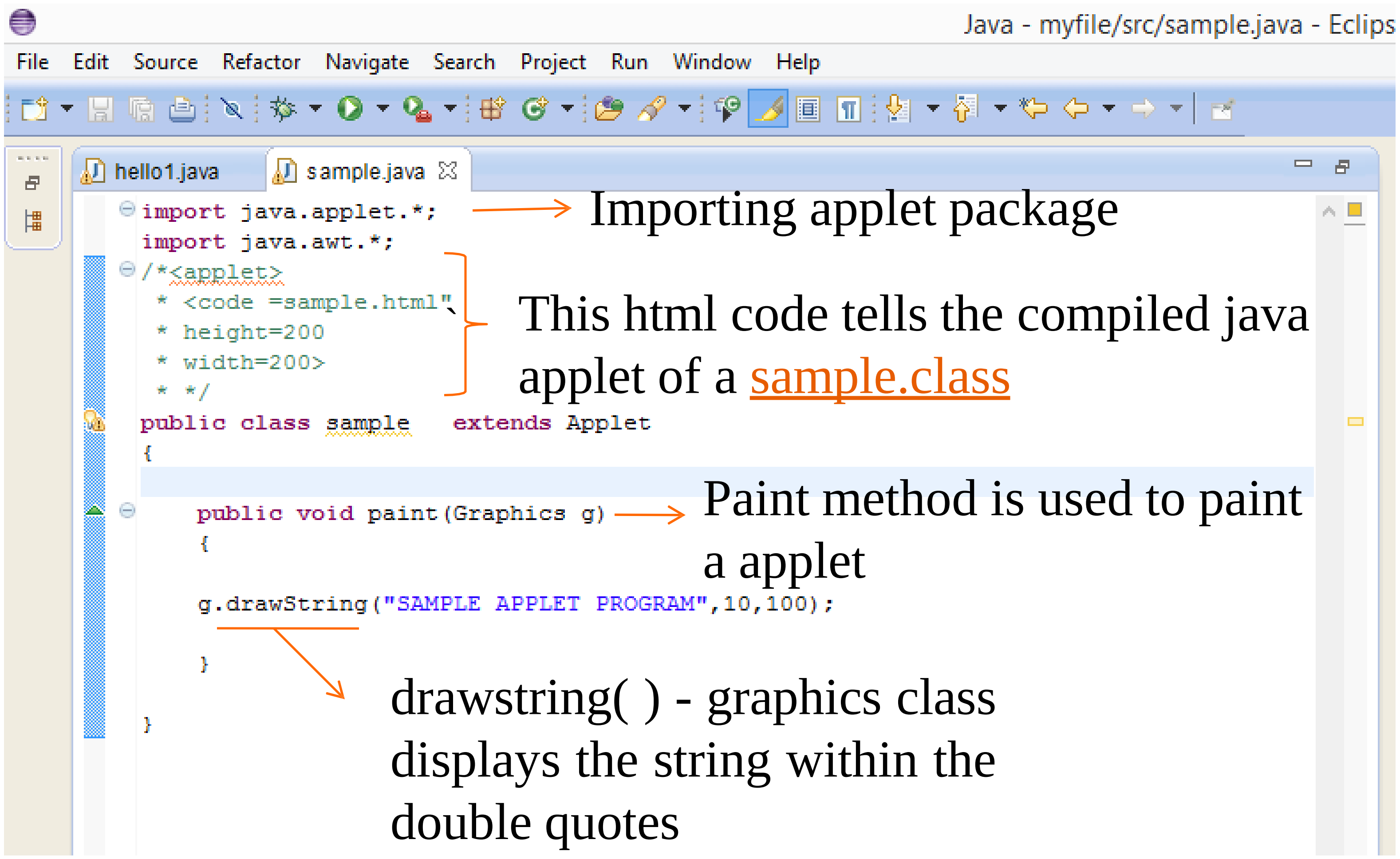
INTRODUCTION

- Main feature in java is creating a graphical interface .
- Graphics in any language gives a wonderful look and feel to the users .
- Two packages that are mainly used to draw graphics.
  - Applet package
  - awt package

APPLET

- An **applet** is a Java program that runs in a Web browser
- An applet is a Java class that extends the **java.applet.Applet** class
- A main() method is not invoked on an applet
- Applets are designed to be embedded within an HTML page.

EXAMPLE APPLET PROGRAM



Java - myfile/src/sample.java - Eclips
File Edit Source Refactor Navigate Search Project Run Window Help
hello1.java
sample.java

```
import java.applet.*;
import java.awt.*;
/*<applet>
 * <code =sample.html"
 * height=200
 * width=200>
 * */
public class sample   extends Applet
{

    public void paint(Graphics g)
    {

    g.drawString("SAMPLE APPLET PROGRAM",10,100);

    }

}
```

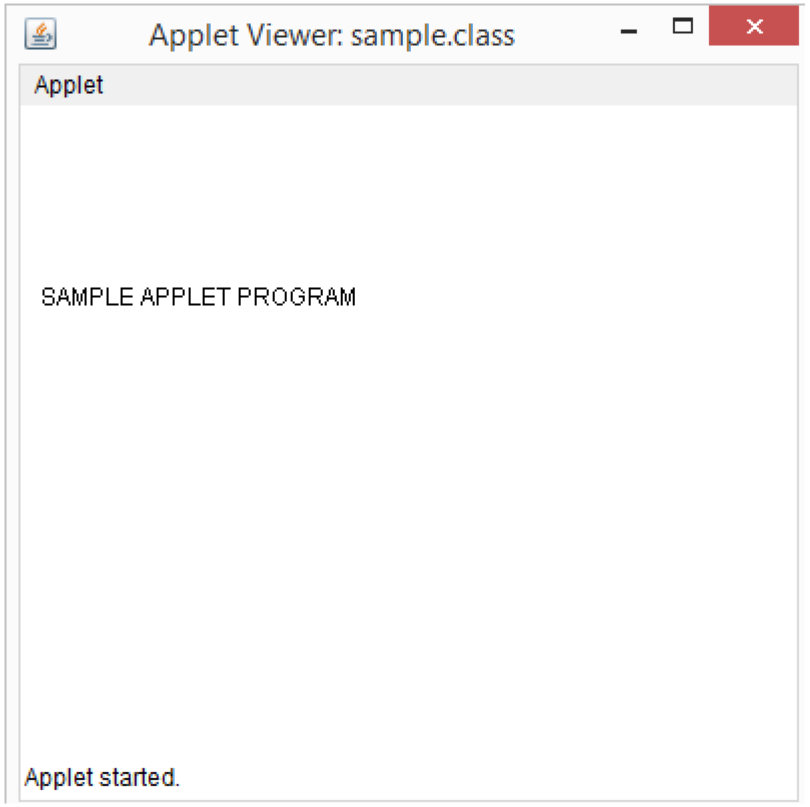
Importing applet package

This html code tells the compiled java applet of a sample.class

Paint method is used to paint a applet

drawstring( ) - graphics class displays the string within the double quotes

OUTPUT



```
SAMPLE APPLET PROGRAM
```

```
Applet started.
```

AWT PACKAGE

- The Abstract Window Toolkit (AWT)
- It is Java's original platform-independent windowing, graphics, and user-interface toolkit.
- The AWT classes are contained in the java.awt package.
- It is used to create a interactive page with buttons , text box and other tools .

Graphics class

- Graphics class include methods for drawing shapes , images to the screen inside your applet
- Graphics class contain several inbuilt methods to create graphical interface.

DRAWING STRING IN APPLET

drawString()

- drawString() is used to display string in Graphical area.

SYNTAX

drawString(String str, int x, int y)

- String to be displayed
- x and y position on the graphical window.

DRAWING LINES

drawLine()

- This method is used to draw a line.

SYNTAX

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

- This method contains two pair of coordinates, (x1, y1) and (x2, y2) as arguments
- draws a line between them.

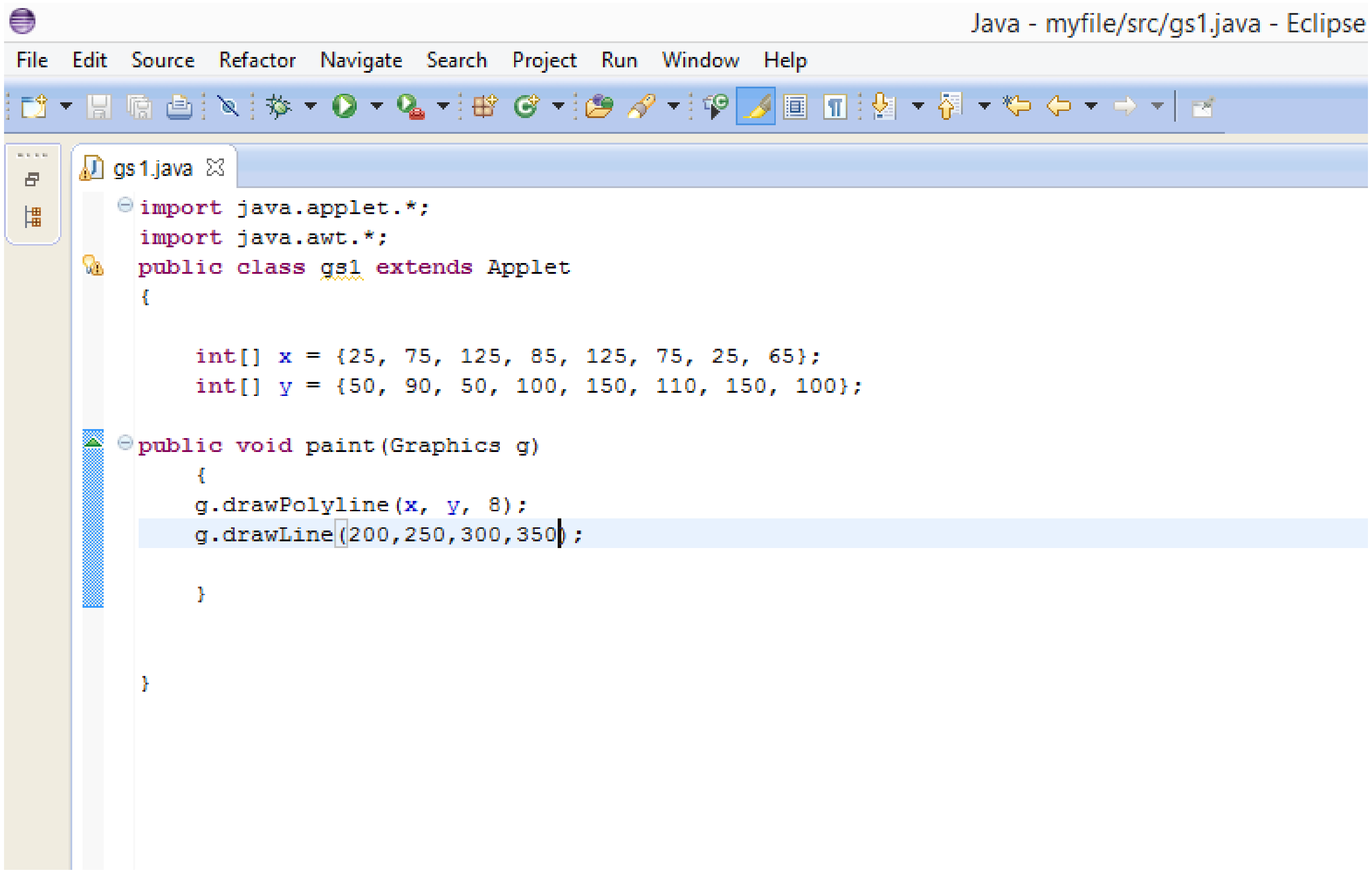
drawPolyline()

- It connects the xPoints and yPoints arrays
- It does not connect the endpoints.

<u>SYNTAX</u>

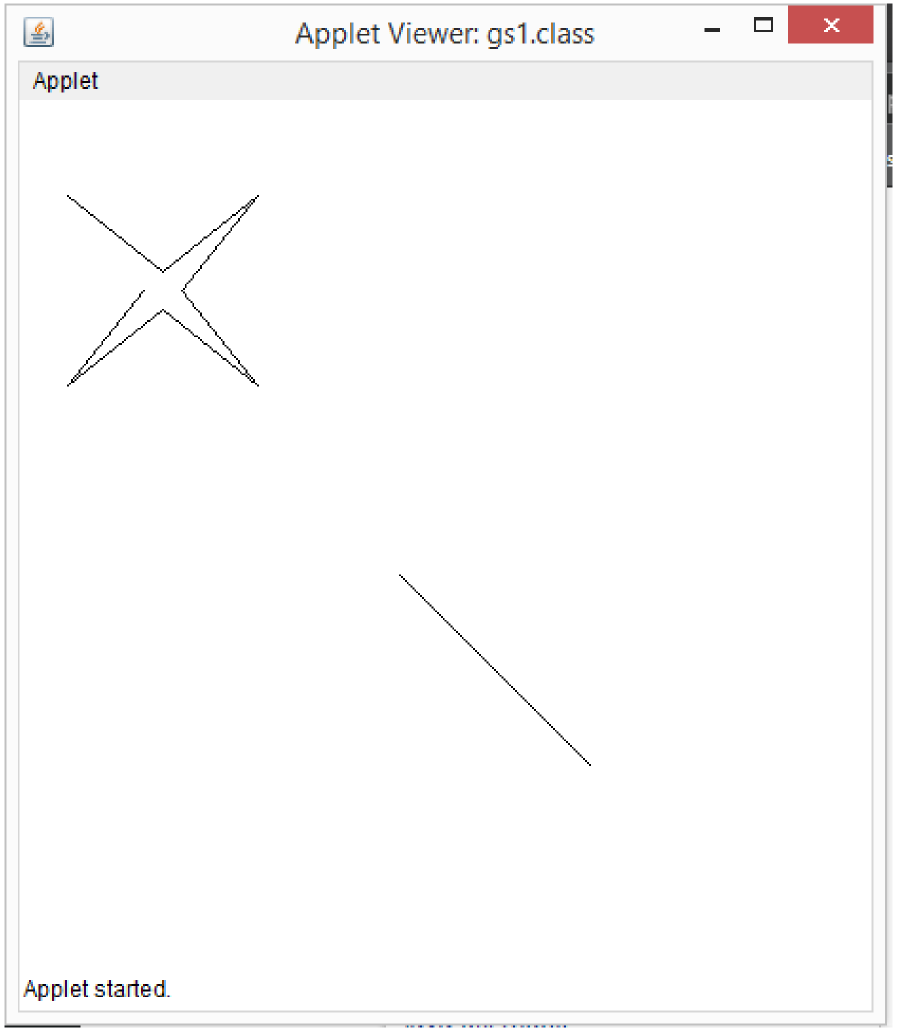
drawPolyline(int[] xPoints,int[] yPoints,int nPoints)

SAMPLE CODE


Java - myfile/src/gs1.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
gs1.java

```
import java.applet.*;
import java.awt.*;
public class gs1 extends Applet
{

    int[] x = {25, 75, 125, 85, 125, 75, 25, 65};
    int[] y = {50, 90, 50, 100, 150, 110, 150, 100};

public void paint(Graphics g)
    {
    g.drawPolyline(x, y, 8);
    g.drawLine(200,250,300,350);

    }


}
```

OUTPUT



Applet started.

DRAWING SHAPE PRIMITIVES

drawRect()

- Used to draw rectangle shape in an applet.

SYNTAX

drawRect(int xTopLeft, int yTopLeft, int width, int height);

- First two points represents **x** and **y** coordinates of the top left corner
- Next two represent the **width** and the **height** of the rectangle.

drawOval()

- Used to draw circle and oval in an applet.

SYNTAX

drawOval(int xTopLeft, int yTopLeft, int width, int height);

- First two arguments represents **x** and **y** coordinates of the top left .
- Third and fourth argument represent the **width** and the **height** of the rectangle .

drawArc()

- Arc is same as oval
- first four are same as arguments of drawOval( )
- Next arguments represents starting angle and degrees around the arc.

<u>**SYNTAX**</u>

drawArc(int xTopLeft, int yTopLeft, int width, int height, int startAngle, int arcAngle);

drawRoundRect()

- By using this method we can draw rounded rectangle

SYNTAX

drawRoundRect(int xTopLeft, int yTopLeft, int width, int height, int arcWidth, int arcHeight)

- Rounded rectangle contains same argument as drawRect().
- In rounded rectangle two extra arguments representing the width and height of the angle of corners.

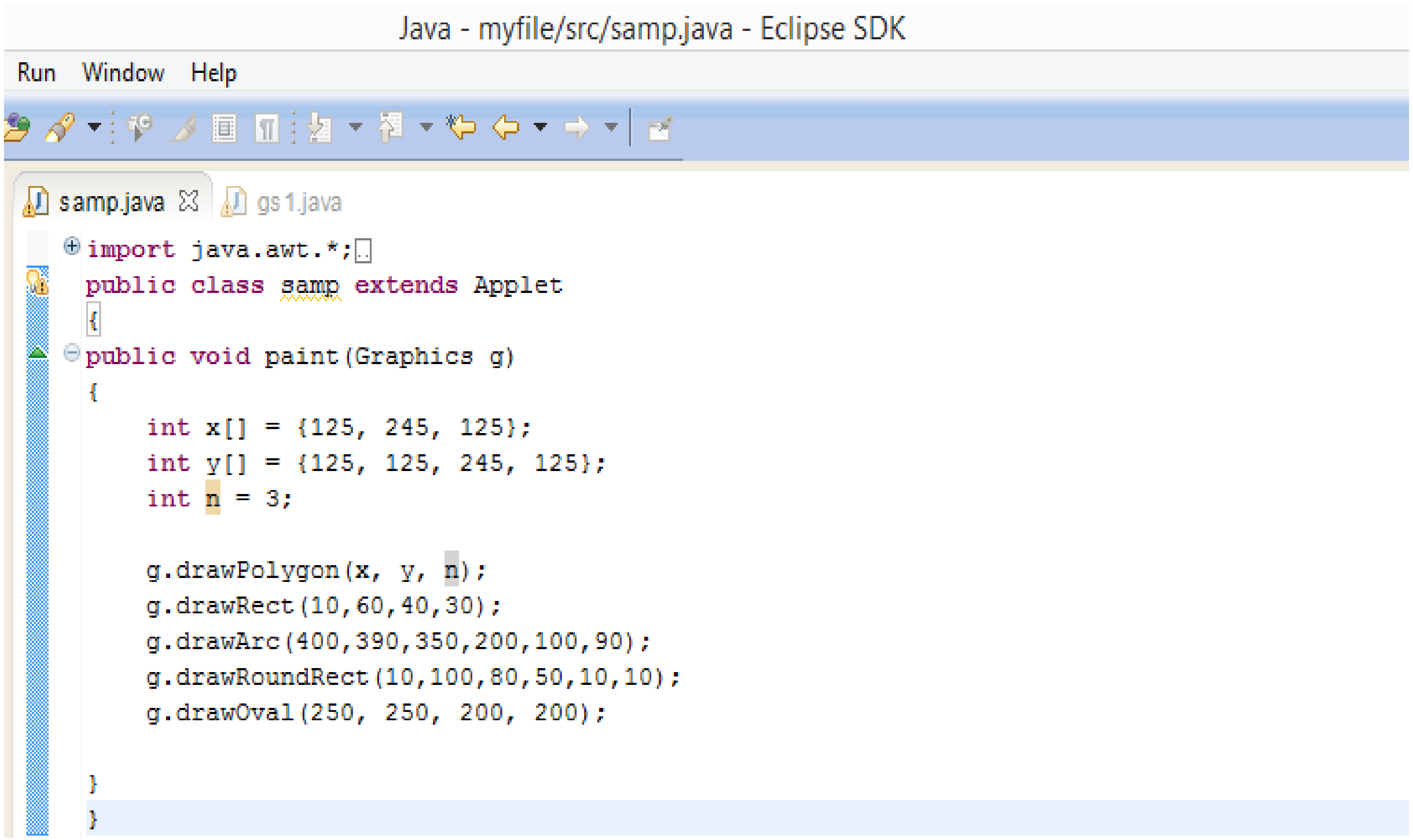
drawPolygon()

- This method Draws an outline polygon as per the coordinates specified in the x[] and y[] arrays
- Numpoints - number of elements in the array

SYNTAX

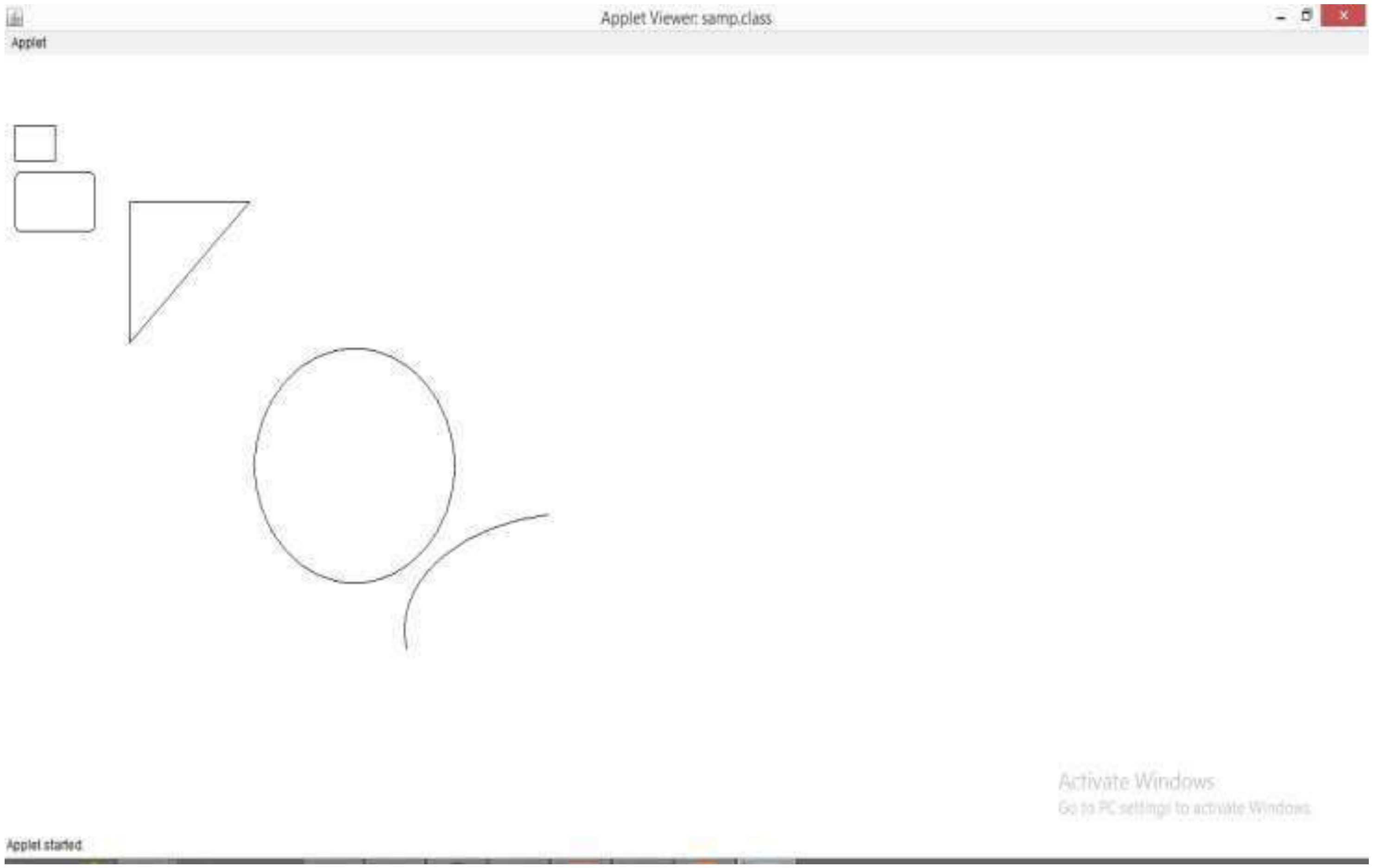
drawPolygon(int[] xPoints, int[] yPoints, int numPoint);

Sample code



```
import java.awt.*;
public class samp extends Applet
{
public void paint(Graphics g)
{
    int x[] = {125, 245, 125};
    int y[] = {125, 125, 245, 125};
    int n = 3;

    g.drawPolygon(x, y, n);
    g.drawRect(10,60,40,30);
    g.drawArc(400,390,350,200,100,90);
    g.drawRoundRect(10,100,80,50,10,10);
    g.drawOval(250, 250, 200, 200);

}
}
```

OUTPUT


Applet Viewer: samp.class
Applet
Applet started.
Activate Windows
Go to PC settings to activate Windows

FILLING PRIMITIVE SHAPES

fillOval()

- The fillOval() method draws a filled oval .
- We can’t specify the oval's center point and radii.
- The filled oval is one pixel smaller to the right and bottom than requested.

SYNTAX

fillOval(int xTopLeft, int yTopLeft, int width, int height);

fillArc()

- The fillArc() method is similar to the drawArc() method except that it draws a filled arc .
- If width and height are equal and arcAngle is 360 degrees
- fillArc() draws a filled circle.

SYNTAX

fillArc(int xTopLeft, int yTopLeft, int width, int height, int startAngle, int arcAngle);

fillRect()

- fillRect() method draws a filled .
- The filled rectangle is one pixel smaller to the right and bottom than requested.
- If width or height is negative, nothing is drawn.

<u>SYNTAX</u>

fillRect(int xTopLeft, int yTopLeft, int width, int height);

fillPolygon()

- The fillPolygon() method draws a polygon.
- If xPoints or yPoints does not have numPoints elements, it throws the run-time exception andIllegalArgumentException.
- If the polygon is not closed, fillPolygon() adds a segment connecting the endpoints.

SYNTAX

fillPolygon(int[] xPoints, int[] yPoints, int numPoint);

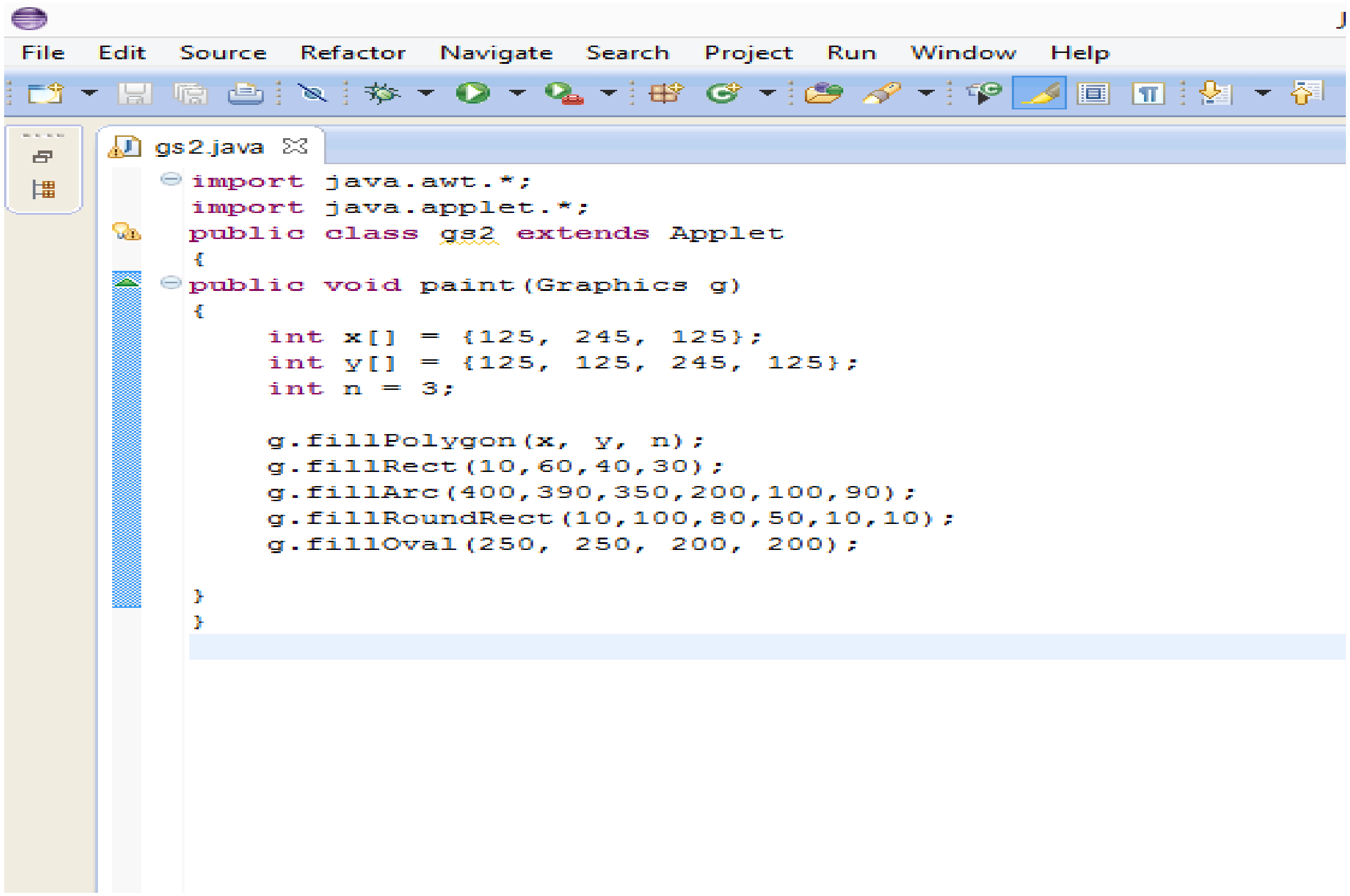
fillRoundRect()

- The fillRoundRect() method is similar to drawRoundRect() method except that it draws a filled rectangle on the drawing area
- If width, height, arcWidth, and arcHeight are all equal, you get a filled circle.

SYNTAX

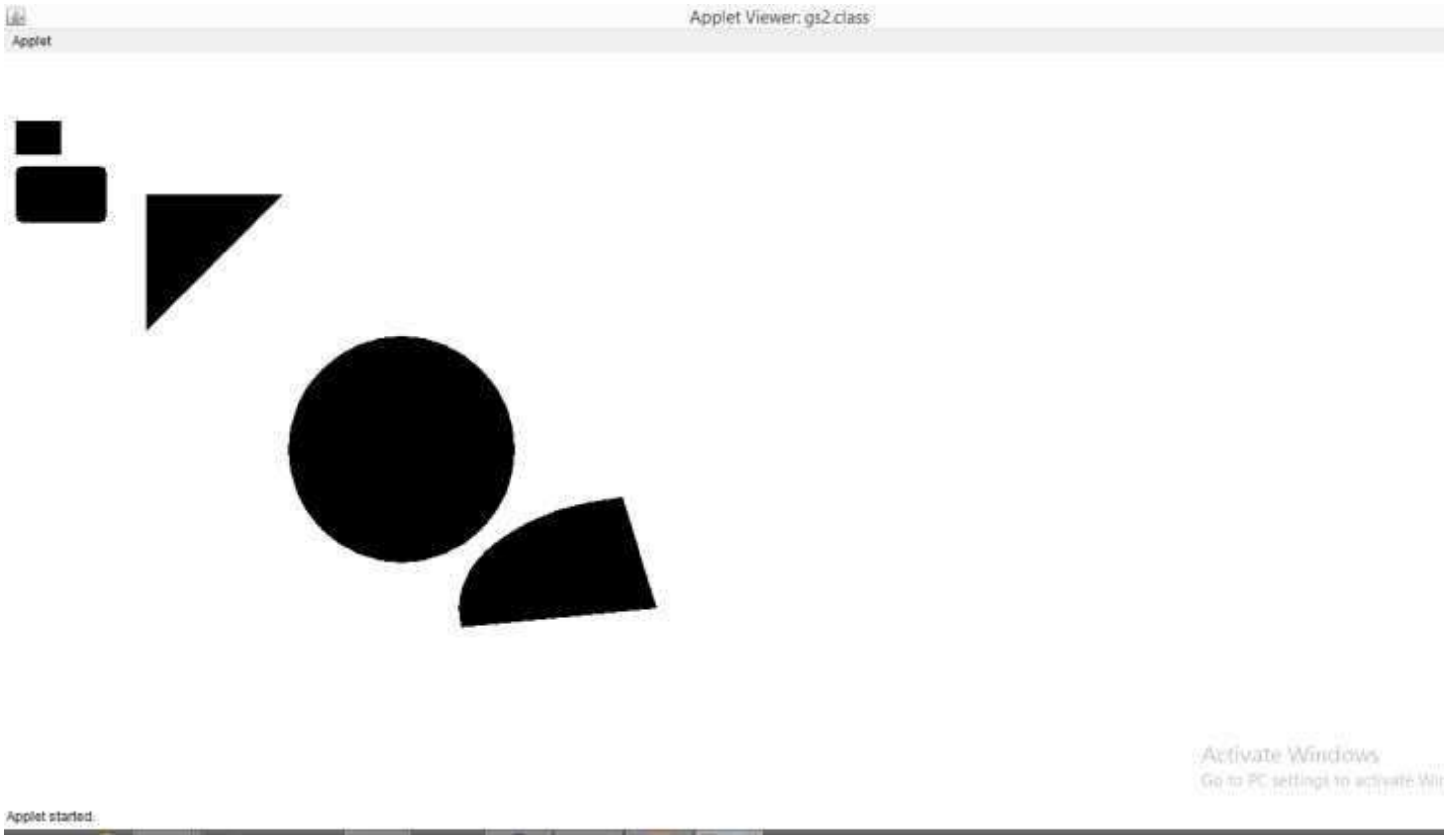
fillRoundRect(int xTopLeft, int yTopLeft, int width, int height, int arcWidth, int arcHeight);

SAMPLE CODE



```
import java.awt.*;
import java.applet.*;
public class gs2 extends Applet
{
public void paint(Graphics g)
{
      int x[] = {125, 245, 125};
      int y[] = {125, 125, 245, 125};
      int n = 3;

      g.fillPolygon(x, y, n);
      g.fillRect(10,60,40,30);
      g.fillArc(400,390,350,200,100,90);
      g.fillRoundRect(10,100,80,50,10,10);
      g.fillOval(250, 250, 200, 200);

}
}
```

OUTPUT


Applet Viewer: gs2.class
Applet
Applet started.
Activate Windows
Go to PC settings to activate Wi

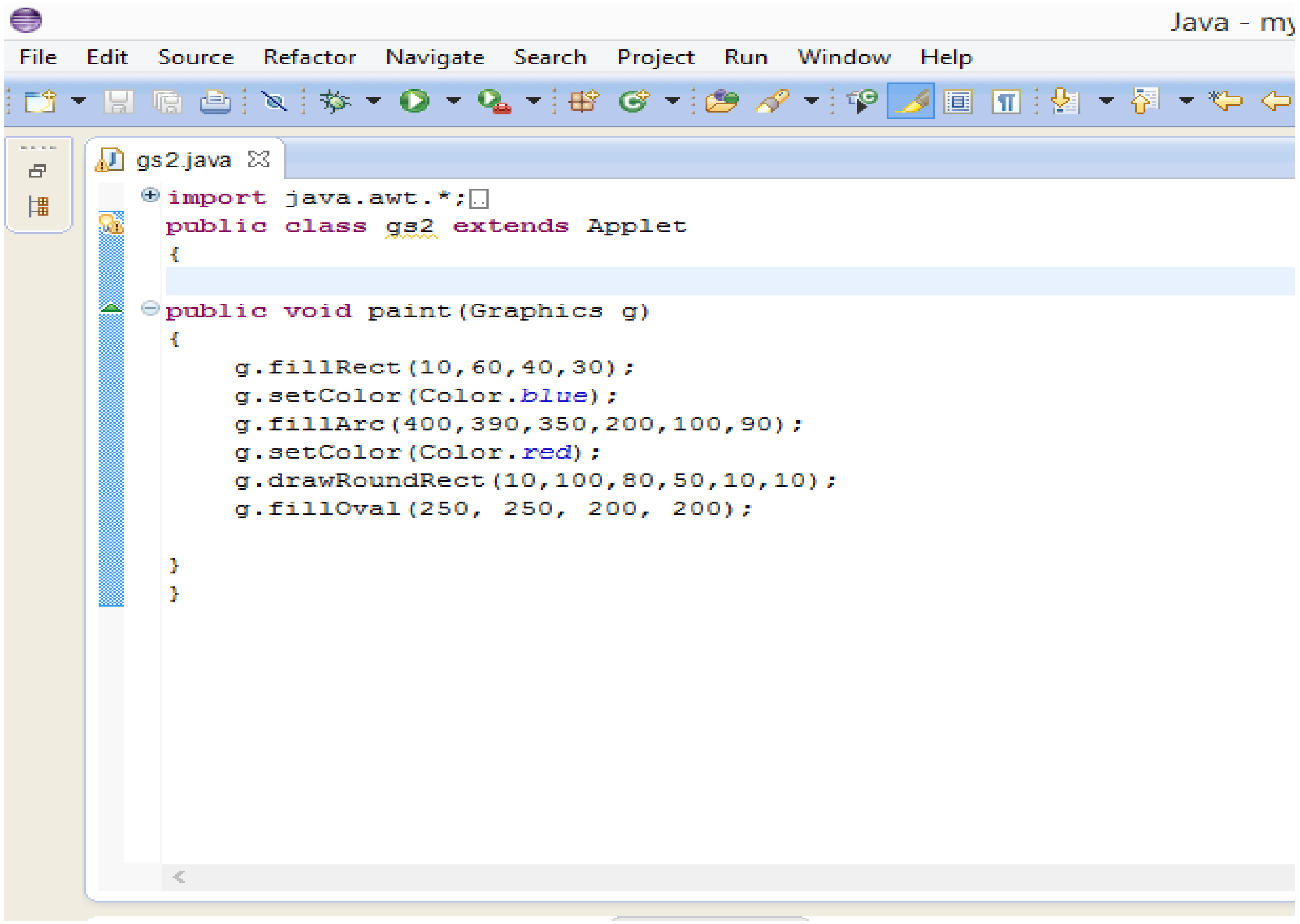
setColor( )

- AWT color class used to specify any color we need.
- color is specified as ***Color.Blue***
- By default, graphics objects are drawn in the current foreground color.
- We can change this color by calling the setColor( ).

EX:

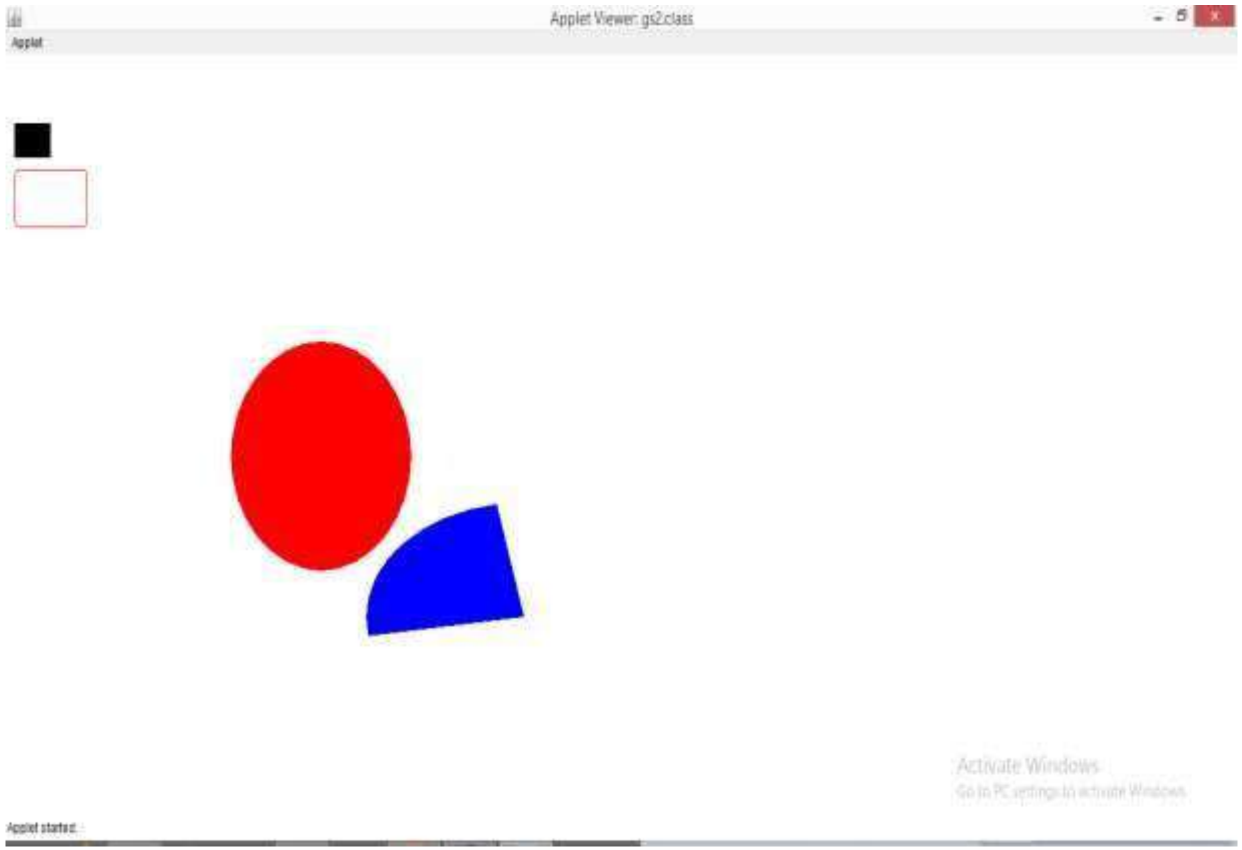
g.setColor(Color . yellow);

Sample code


Java - my
File Edit Source Refactor Navigate Search Project Run Window Help
gs2.java

```
import java.awt.*;
public class gs2 extends Applet
{

public void paint(Graphics g)
{
     g.fillRect(10,60,40,30);
     g.setColor(Color.blue);
     g.fillArc(400,390,350,200,100,90);
     g.setColor(Color.red);
     g.drawRoundRect(10,100,80,50,10,10);
     g.fillOval(250, 250, 200, 200);

}
}
```

OUTPUT


Applet Viewer: gs2.class
Applet
Applet started.
Activate Windows
Go to PC settings to activate Windows

THANK YOU