Chapter - 7

Applet Programming

Applet and Application

- An applet is a Java program that runs on a web page
  - Applets can be run within any modern browser
  - To run modern Java applets, old browsers need an up-to-date Java plugin
  - appletviewer is a program that can run
  - In Java, non-applet programs are called applications.

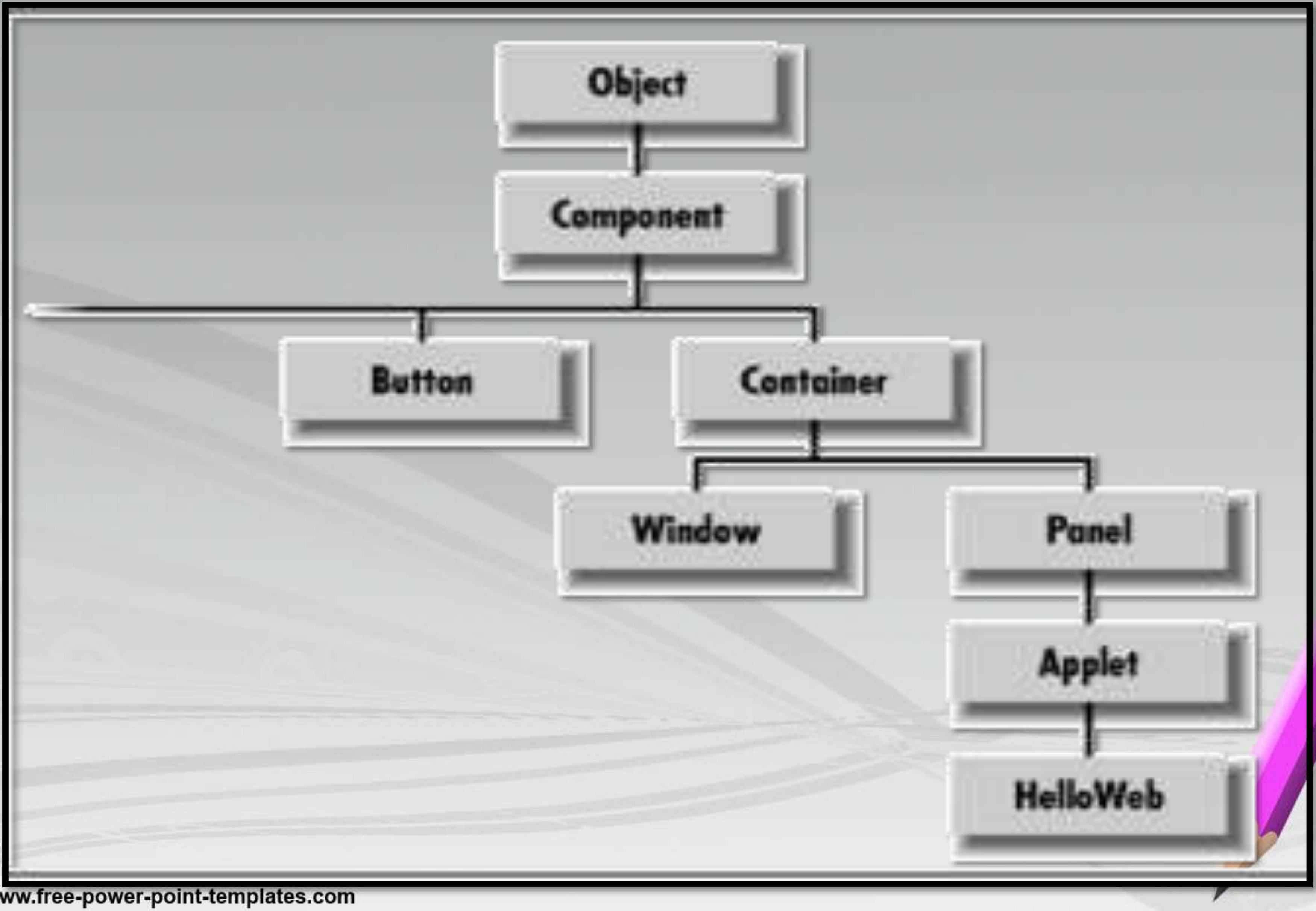
Applet…

- **Advantage of Applet:**
  - – 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.

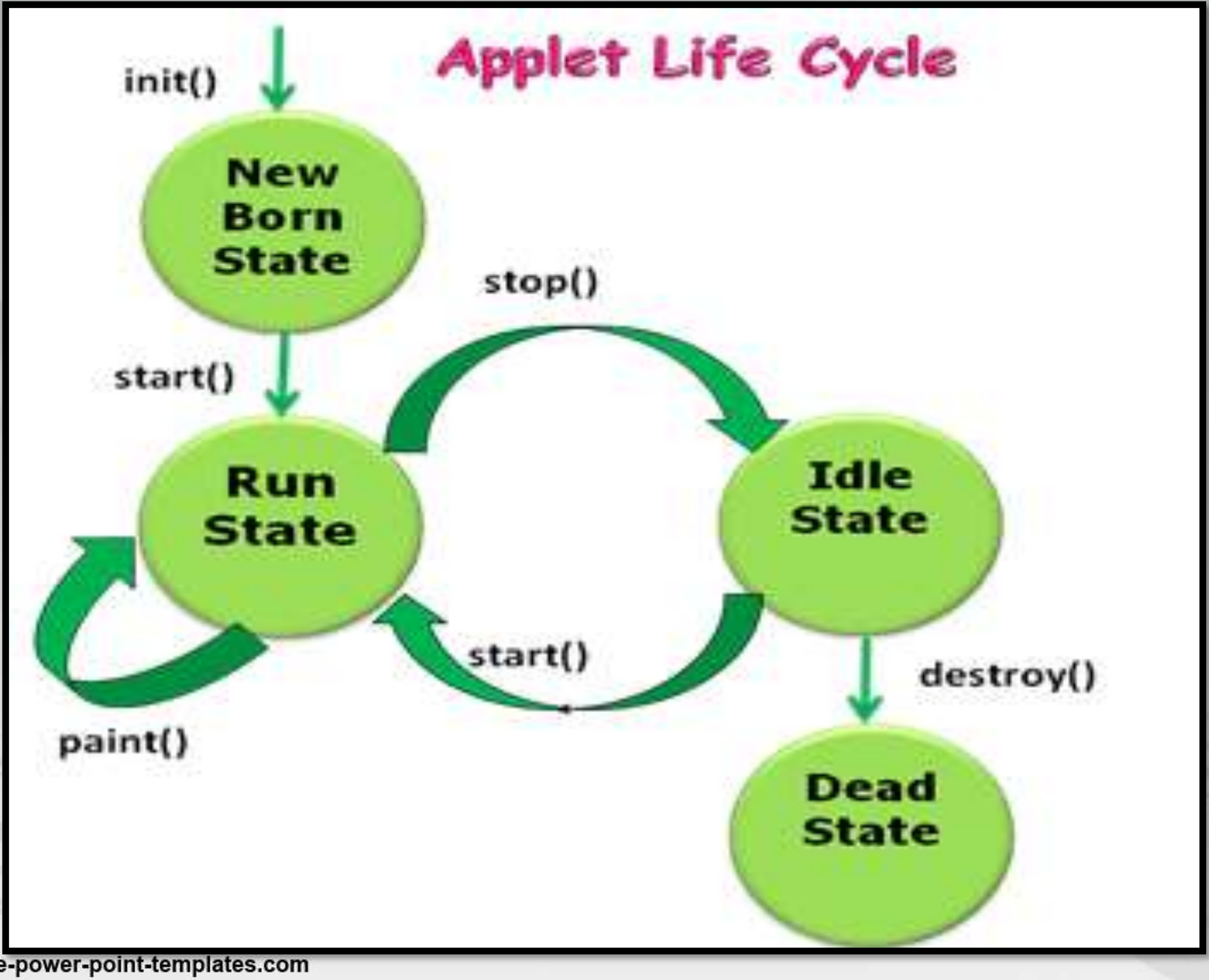
Application vs. Applet

- **Application**
  - Trusted (i.e., has full access to system resources)
  - Invoked by Java Virtual Machine (JVM, java), e.g.,
  - java HelloWorld
  - Should contain a main method, i.e.,
  - public static void main(String[])
- **Applet**
  - Not trusted (i.e., has limited access to system resource to prevent security breaches)
  - Invoked automatically by the web browser
  - Should be a subclass of class java.applet.Applet

Hierarchy of Applet class



Life Cycle of Applet



Life Cycle…

- Following methods are executed during life cycle of an applet.
  - init() :
    - It is used to initialized the Applet.
    - It is invoked when applet is first loaded.
    - It is invoked only once.
  - start() :
    - It is invoked after the init() method.
    - It is invoked when an applet is started or restarted.

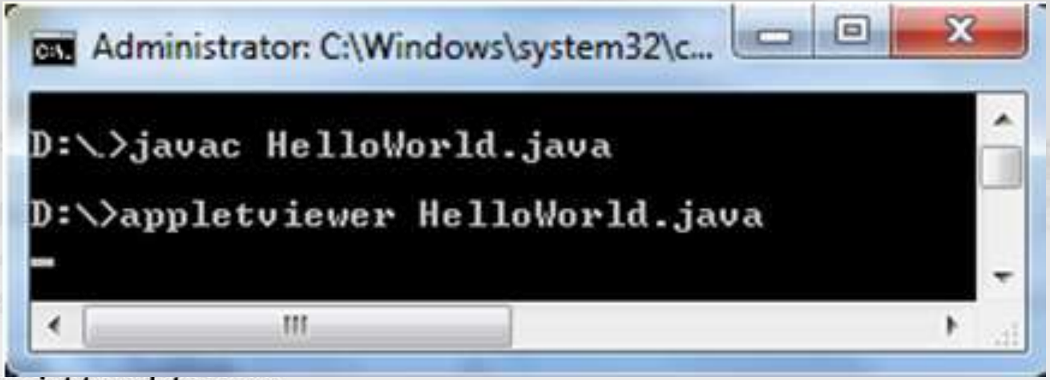
Life Cycle…

- paint() :
  - This method invoked immediately after the start() method.
  - It also called at any time when the applet needs to repaint itself in the browser.
- stop() :
  - It is used to stop the Applet.
  - It is invoked when Applet is stop or browser is minimized.
- destroy() :
  - It is used to destroy the Applet.
  - It is invoked only once.
  - It is invoked when the browser unloads the applet.

Simple Applet Program

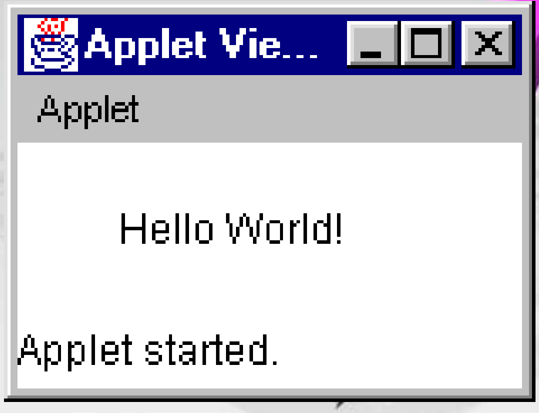
```
import java.awt.*;
import java.applet.*;
/* <applet code="HelloWorld" width=200 height=100>
    </applet> */
public class HelloWorld extends Applet
 {    public void paint( Graphics g )
   {
        g.drawString( "Hello World!", 30, 30 );
    }
}
```

Example of Life Cycle



```
D:\>javac HelloWorld.java

D:\>appletviewer HelloWorld.java
_
```



```
Hello World!

Applet started.
```

<Applet> tag parameters

- The HTML applet tag contains the following parameters:
  - <Applet code="name of .class file"
  - codebase="URL where code is loaded from"
  - name="applet identifier"
  - align="LEFT|RIGHT|CENTER
  - width="size in pixels"
  - height="size in pixels“>
  - <param name=“aName1” value=“aValue”>
  - <param name=“aName2” value=“aValue”>
  - </Applet>

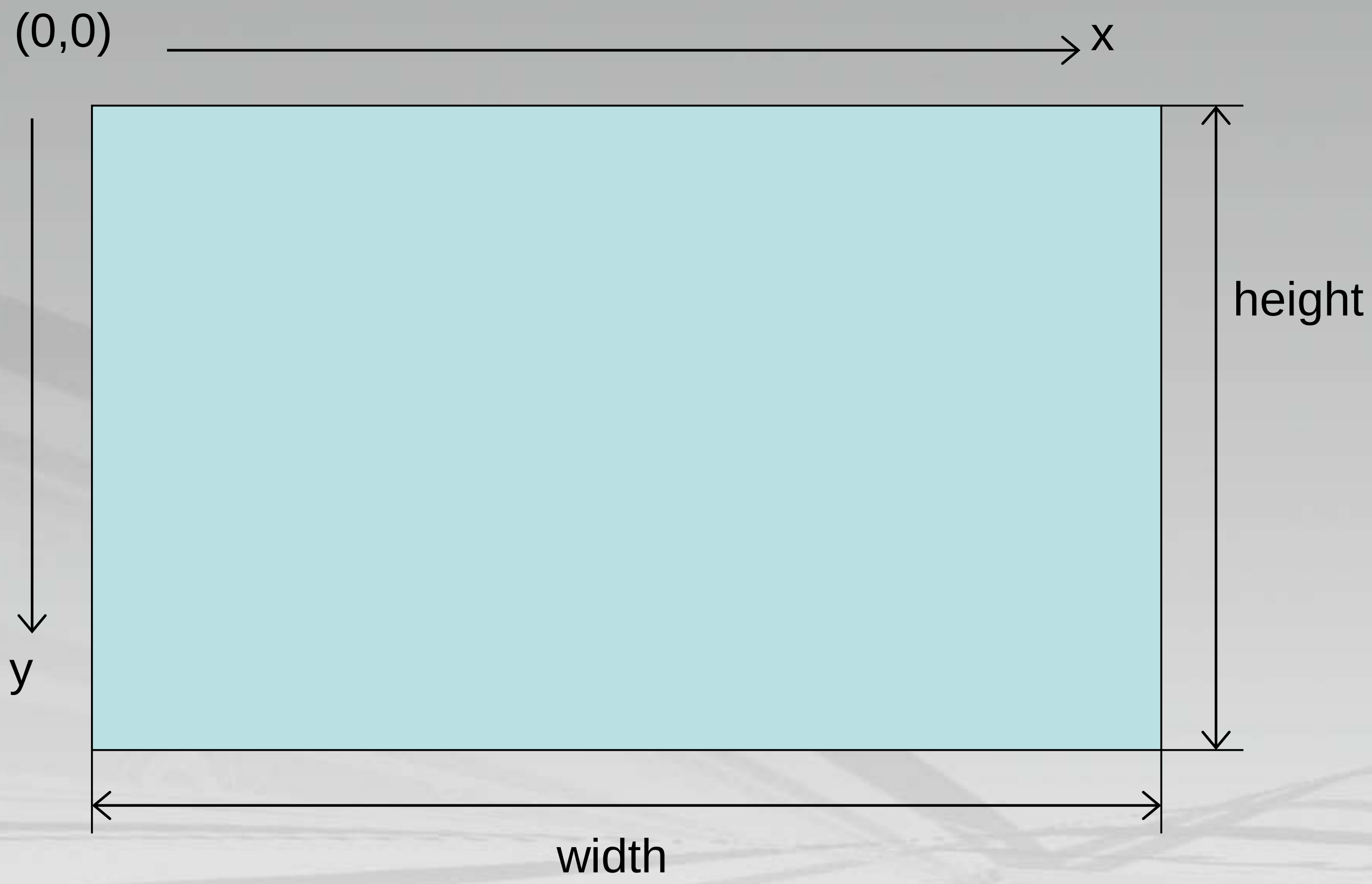
Embedding Applet into HTML

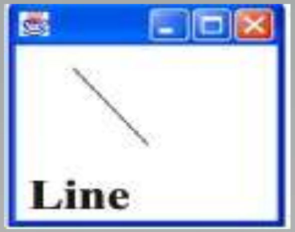
- <html>
- <head>
- <title>HTML applet Tag</title>
- </head> <body>
- <body>
- <applet code="newClass.class" width="300" height="200"> </applet>
- </body>
- </html>

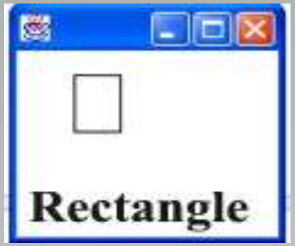
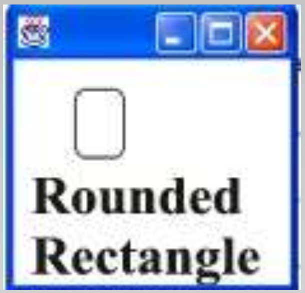
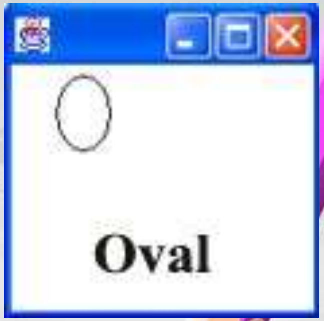
Graphics Class

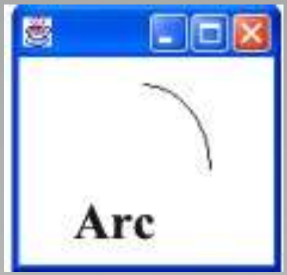
- Graphics class belongs into java.awt package.
- Methods:
  - g.drawString(“Hello”, 20, 20); Hello
  - g.drawRect(x, y, width, height); □
  - g.fillRect(x, y, width, height); ■
  - g.drawOval(x, y, width, height); ○
  - g.fillOval(x, y, width, height); ●
  - g.setColor(Color.red); ✸

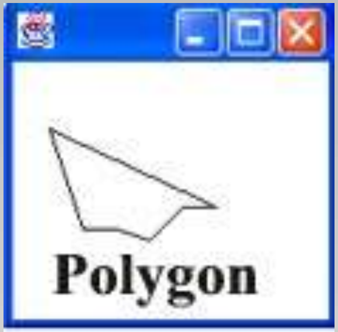
Graphics Coordinate

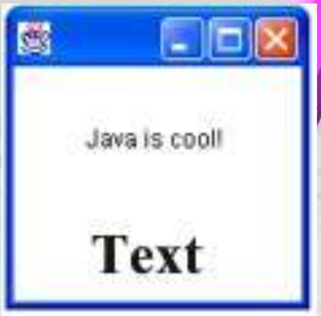


Line	g.drawLine(35, 45, 75, 95);
	drawLine(int x1, int y1, int x2, int y2)
	Used to draw a straight line from point (x1,y1) to (x2,y2).
		

Rectangle	g.drawRect(35, 45, 25, 35); drawRect(int x, int y, int width, int length) Used to draw a rectangle with the upper left corner at (x,y) and with the specified width and length.	
Round Edge Rectangle	g.drawRoundRect(35,45,25,35,10,10); drawRoundRect(int x, int y, int width, int length, int arcWidth, int arcHeight) Used to draw a rounded edged rectangle. The amount of rounding is controlled by arcWidth and arcHeight.	
Oval / Circle	g.drawOval(25, 35, 25, 35); g.drawOval(25, 35, 25, 25); → circle drawOval(int x, int y, int width, int length) Used to draw an oval inside an imaginary rectangle whose upper left corner is at (x,y). To draw a circle keep the width and length the same.	

Arc	g.drawArc(35, 45, 75, 95, 0, 90); drawArc(int x, int y, int width, int length, int startAngle, int arcAngle) Used to draw an arc inside an imaginary rectangle whose upper left corner is at (x,y).	

Polygon	int [ ] x = {20, 35, 50, 65, 80, 95};
	int [ ] y = {60, 105, 105, 110, 95, 95};
	g.drawPolygon(x, y, 6);
	drawPolygon(int x[ ], int y[ ], int n)
	Used to draw a polygon created by n line segments. The command will close the polygon. (x-coordinates go in one array with accompanying y-coordinates in the other)
		

String
(text)
	g.drawString("Java is cool!", 40, 70);
	drawString(String str, int x, int y);
	Draws a string starting at the point indicated by (x,y). Be sure you leave enough room from the top of the screen for the size of the font.
		 Java is cool!

The java.awt.Color Class

- Instances of the Color class represent colors.
- new Color(r, g, b)
  - where r, g, b are the values of the red, green, and blue components, respectively. They are in the in the range of 0 to 255.
- Predefined constants
  - BLACK ORANGE YELLOW BLUE GREEN PINK CYAN LIGHTGRAY RED DARKGRAY MAGENTA WHITE

The java.awt.Font Class

- Fonts are specified with three attributes:
  - font name: Serif Sans-serif ,Monospaced Times New Roman, Helvetica, Courier etc.
  - font style: PLAIN,BOLD,ITALIC
    - Styles can be combined: Font.BOLD|Font.ITALIC
  - font size: a positive integer
- A font can be created as follows:
- new Font(name, style, size)

Example…

```
import java.awt.*;
import java.applet.*;
/* applet code=“Test_Graphics” width=300 height=200></applet> */
public class Test_Graphics extends Applet
{
   public void paint(Graphics g)
    {
      Dimension d = getSize();
      g.setColor(Color.BLACK);
      g.fillRect(0, 0, d.width, d.height); // paint background
      g.setFont(new Font("San-serif", Font.BOLD, 24));
      g.setColor(new Color(255, 215,0));
      g.drawString("Hello, world!", 60, 40);
      g.drawImage(getImage(getCodeBase(), “Rabbit.jpg"),20, 60, this);
   }
}
```

Thank You