

# Java Programming

(Very) Primitive Graphics

## Agenda

- Brief Review
- JavaDoc
- Lecture
  - Objective: Overview of how to use already made objects
  - Objective: Learn about primitive Graphics functionality in Java

## From previous experience

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

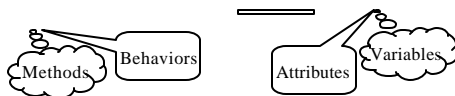
public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello World", 50, 25);
    }
}
```

## Observations on applet

- Unlike application, where you control everything from `main()`, applet has a strict framework which you must follow
  - ie. Change `paint()` to a different name and the applet will compile but won't work.
  - ie. `g.drawString(msg, x, y)` works only from inside `paint()` method since `Graphics g` is available to `paint()` only.

## What is an object?

- An object is a software bundle of data and code.

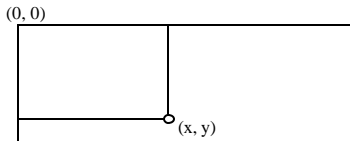


## Back to Graphics

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

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello World", 50, 25);
    }
}
```

## Java graphic coordinate System



- Units measure in pixels; start from 0, 0
- Notes: all subsequent Graphics sample code assume execution inside `public void paint(Graphics g)`

## Color Control

- `g.setColor(new Color( red, green, blue));`
  - `g.setColor(new Color( 255, 255, 255)); // white`
- `g.setColor( Color.blue );`
  - Common used colors are predefined

| Color constant             | Color   | RGB value     |
|----------------------------|---------|---------------|
| <code>Color orange</code>  | orange  | 255, 200, 0   |
| <code>Color pink</code>    | pink    | 255, 175, 175 |
| <code>Color cyan</code>    | cyan    | 0, 255, 255   |
| <code>Color magenta</code> | magenta | 255, 0, 255   |
| <code>Color yellow</code>  | yellow  | 255, 255, 0   |
| <code>Color black</code>   | black   | 0, 0, 0       |
| <code>Color white</code>   | white   | 255, 255, 255 |

## Font Control

- `g.setFont(new Font("font name", style, size));`
  - `g.setFont(new Font("Courier", Font.ITALIC, 24));`
  - `g.setFont(new Font("TimesRoman", Font.BOLD, 12));`
- Available fonts are varied from system to system
- All systems have TimesRoman, Courier, Helvetica, & Dialog

## Drawing Lines

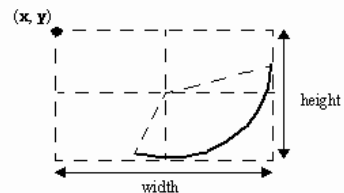
- `g.drawLine(x1, y1, x2, y2);`
- What would `g.drawLine(10, 20, 100, 20)` draw?
- What would `g.drawLine(100, 20, 100, 70)` draw?
- What would `g.drawLine(10, 20, 100, 70)` draw?

## Rectangles

- `g.drawRect(x, y, width, height);`
- `g.drawRect(10, 15, 100, 100); // square`
- Variations:
  - `g.fillRect(x, y, width, height); // fill with current color`
  - `g.clearRect(x, y, width, height); // fill background`
- More variations:
  - `g.drawRoundRect(x, y, width, height, arc width, arc height) // for rounded corner rectangle`
  - `g.draw3DRect(...)`

## Drawing Arcs

- `g.drawArc( x, y, width, height, startAngle, sweepingArcAngle);`



## Polygons

```
Polygon myTriangle; // declare variable as a Polygon object
myTriangle = new Polygon(); // instantiate a new Polygon object

myTriangle.addPoint(240, 50); // specify the 1st point
myTriangle.addPoint(260, 70); // 2nd point
myTriangle.addPoint(250, 90); // 3rd point

g.drawPolygon(myTriangle); // actually draw it
```

# Break

## Group Assignment

- Group of 3 students (by alphabetical order)
- Follow the guide lines in Assignment 3
- Turn in 1 print out with names of all members and show the instructor the running program on screen
- First groups: 100 points, other groups which complete by 9/18: 95 points.