

Review Unit 1

Graphics, Applets, Methods

The Graphics Shapes

```
public void drawLine(int x1, int y1, int x2, int y2)
```

```
public void fillRect(int x, int y, int width, int height)
```

```
public void drawOval(int x, int y, int width, int height)
```

```
int xPts[]= {x1, x2, x3 ...};
```

```
int yPts[]= {y1, y2, y3 ...};
```

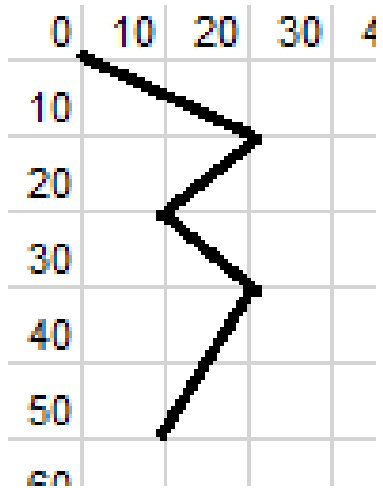
```
public void drawPolygon(int[] xPts, int[] yPts, int nPts)
```

Where x+ goes for
the graphics
methods

Where y+ goes for
the graphics
methods

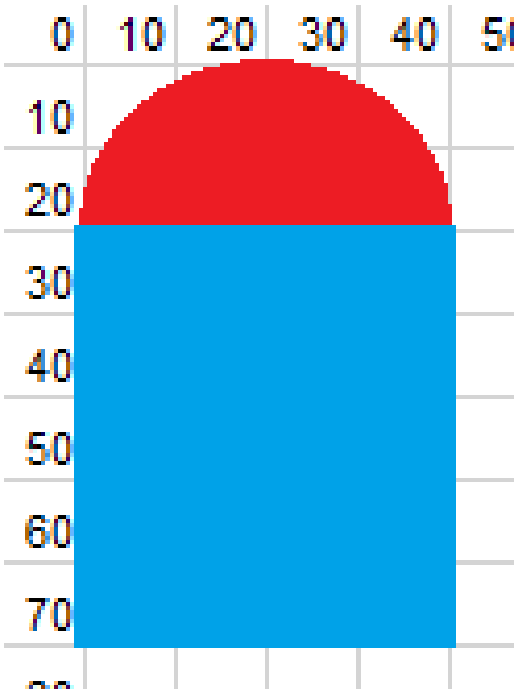
No changes for the
graphics methods

An Example of a Graphics Method



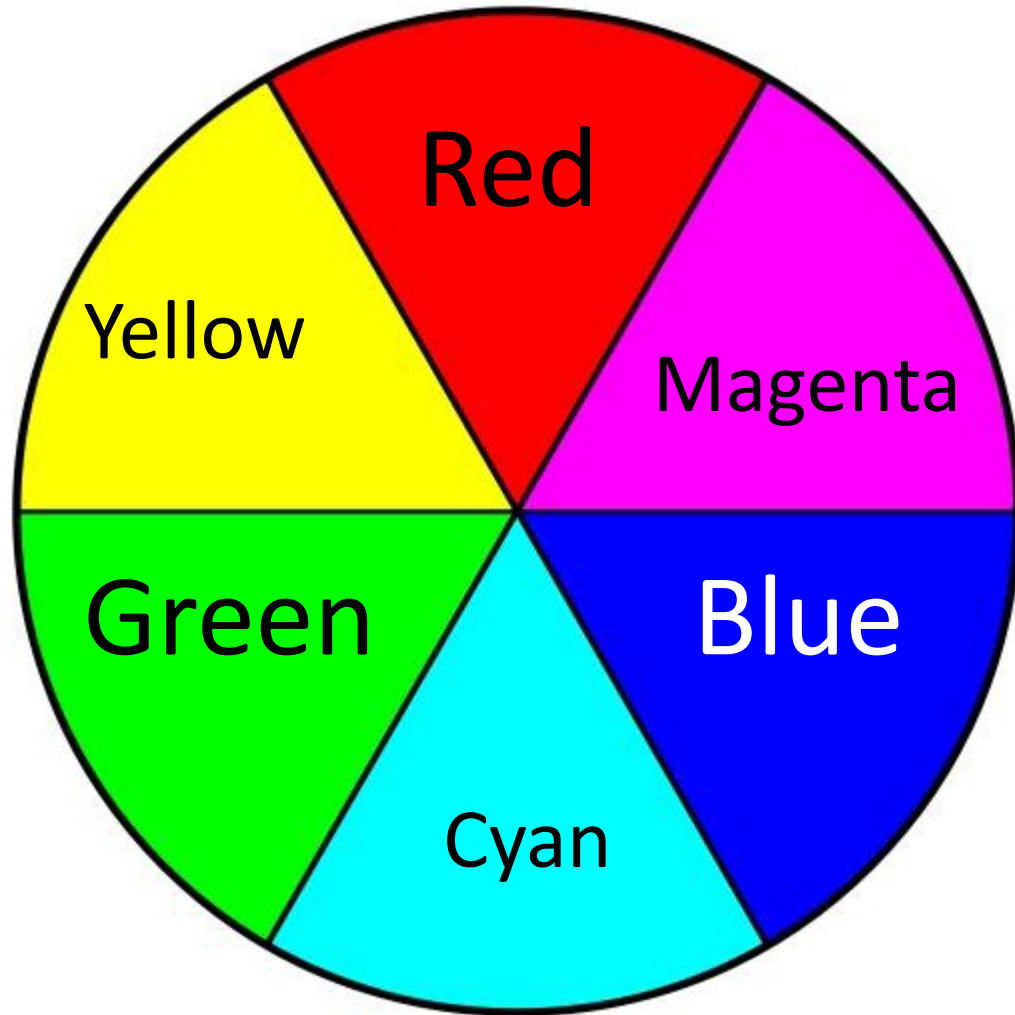
```
public void squiggle (int x, int y)
{
    Graphics g = getGraphics ();
    g.drawLine (x+0, y+0, x+20, y+10);
    g.drawLine (x+20, y+10, x+10, y+20);
    g.drawLine (x+10, y+20, x+20, y+30);
    g.drawLine (x+20, y+30, x+10, y+50);
}
```

Another Example of a Graphics Method



```
public void silo (int x, int y)
{
    Graphics g = getGraphics ();
    g.setColor (Color.red);
    g.fillOval (x + 0, y + 0, 40, 40);
    g.setColor (Color.blue);
    g.fillRect (x + 0, y + 20, 40, 50);
}
```

RGB Colours



$$R + G = Y$$

$$G + B = C$$

$$R + B = M$$

How do you make custom colours?

	R	G	B
red	255	0	0
yellow	255	255	0
green	0	255	0
cyan	0	255	255
blue	0	0	255
magenta	255	0	255
black	0	0	0
white	255	255	255


```
title.setBackground(new Color(255, 0, 0));  
title.setBackground(new Color(255, 255, 0));  
title.setBackground(new Color(0, 255, 0));  
title.setBackground(new Color(0, 255, 255));  
title.setBackground(new Color(0, 0, 255));  
title.setBackground(new Color(255, 0, 255));  
title.setBackground(new Color(0, 0, 0));  
title.setBackground(new Color(255, 255, 255));
```

Parts of the Method

1. Method signature
2. Return type
3. Method name
4. Parameter
5. Parameter name
6. Parameter type

Method Structure

```
//Complex method  
public returnType methodName (paramType paramName) {  
    code;  
    return something;  
}
```

Why are
methods
useful?

Organization

Reusability

Abstraction

Testing

Extensibility

Organization

Breaks things up into smaller logical units.
Think: Tasks in your RPG from last year.

Reusability

Instead of copy/pasting code, call the method.
Think: Less chance of copy/paste errors.

Abstraction

To use someone else's code, call it using the method signature. You don't need to understand the details.
Think: IO

Testing

When we don't repeat code, there are fewer lines for white box testing.
Think: Testing the if 5 times!

Extensibility

By putting code in one place, changes are easier.
Think: Fixing things in seven places instead of one.

Black Box Testing

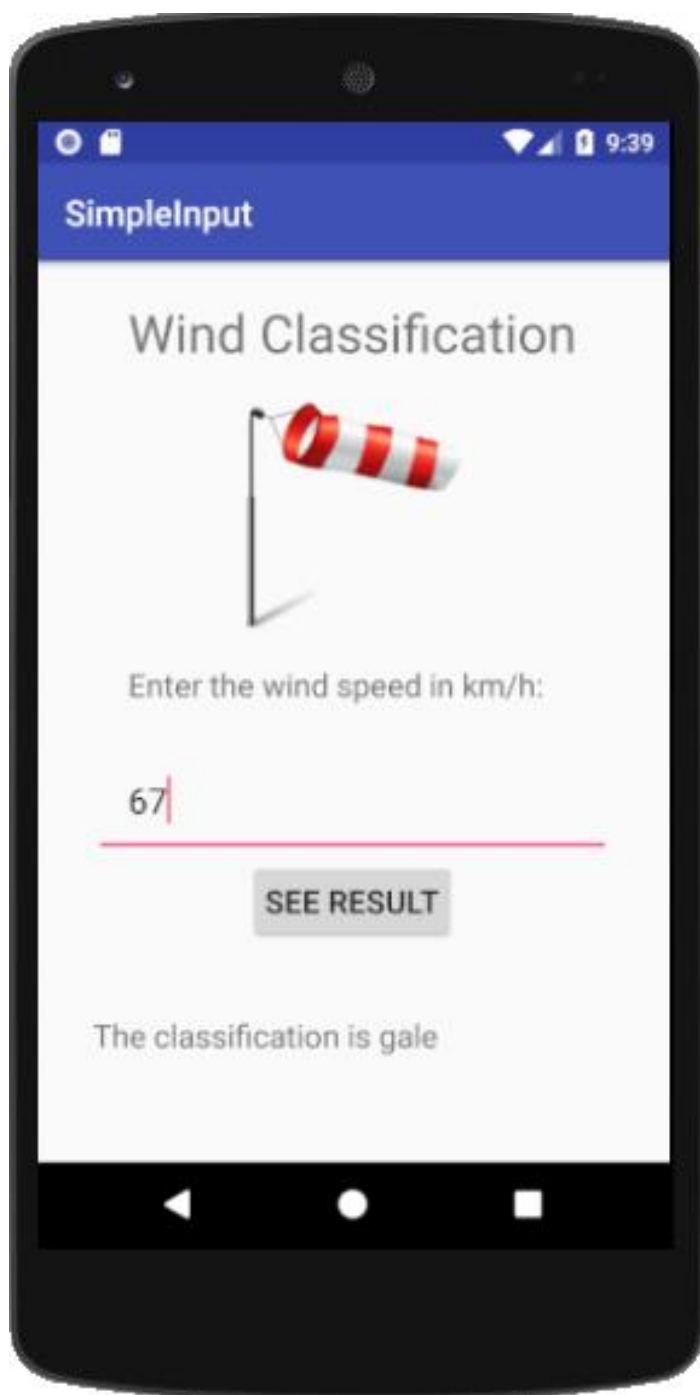
- Testing without looking at the code.
- Only know what is required and what should be produced
- Four kinds of data are tested:
 - (1) Small data
 - (2) Large data
 - (3) Average (normal) data
 - (4) Boundary cases – strange data, odd cases that aren't normally considered

White Box Testing

- Testing that occurs after looking at the code
- Goal is to run every line of code at least once.
- Two major considerations:
 - (1) Ifs
 - Test every clause of the if
 - (2) Loops
 - Avoid the loop
 - Run the loop once
 - Run the loop many times.

Product Development Life Cycle

1. **Analysis** – Proposal for Solution to Problem
2. **Design** – Detailed Plan, Planning Diagrams
3. **Code** – Complete program, **fully tested**, commented.
4. **Reflection** – Direction for next version, sales, advertising, support for current version.



Short
number?

23

Long
number?

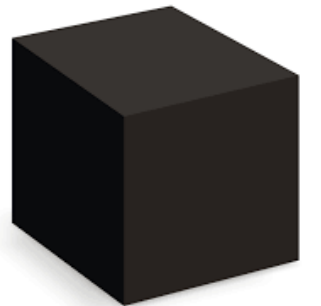
1,234,456,789

Average
number?

134

Boundary-
case
number?

-1, 0, 1



Ariane 5 rocket

- Unmanned rocket with satellites
- A decade of development cost \$7 billion
- Rocket and cargo valued at \$500 million
- Launched June 4, 1996
- Exploded 40 sec after take-off
- Didn't Black Box test "Large Values" for the speed variable. It overloaded memory.

American Megatrends

Cancer Treatment

- Device delivered radiation doses to cancer patients
- Used in Panama City, August 2000 to March 2001
- Didn't white box test "Run Loop Multiple Times"
- It treated multiple doses of radiation as only one dose.
- 28 people receive over doses. 17 die, 11 severely injured.

1. JButton enter = **new** JButton ("Enter");
2. enter.**setBackground** (Color.black);
3. enter.**setForeground** (Color.white);
4. enter.**setPreferredSize** (new Dimension (128, 128));
5. enter.**setFont** (new Font ("Arial", Font.BOLD, 40));
6. enter.**addActionListener** (this);
7. enter.**setActionCommand** ("enter");
8. **add** (enter);
9. enter.**setIcon**(createImageIcon("name.jpg"));
10. enter.**setText**("hi");

Compare and Contrast ActionPerformed and Init

	Init	ActionPerformed
How Often It Runs	Once	The user decides.
When it is run	When the applet begins	When the user clicks a button
General Purpose	To set up the screen	To respond to a user's request
Lines of Code Found there	Declaration Construction (new) Mutation (set) Add addActionListener setActionCommand	Mutation (set) Accessors (get) showStatus
Lines of Code NOT found there	Accessors (get) – you can, it is just pointless.	Construction (new) Add

Terms

drawLine

drawOval

fillLine

fillPolygon

setColor

red

magenta

green

blue

yellow

Cyan

paint

JLabel

JTextField

JButton

Ariane5

Panama

AmericanMegatrends

Forty

Billion

boundary

white

black

init

actionPerformed

getText

organization

abstraction

extensibility

testing

reusability

parameter

methodsignature

returntype

coding