Graphics and Methods

COMP163

"The best programmers are not marginally better than merely good ones. They are an order-of-magnitude better, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability."

Randall E. Stross

First Exam

- The first exam in COMP163 will be in lecture on Monday, September 17
- The exam will cover everything since the beginning of the semester
- You are allowed one 8½ by 11 inch page of notes
- Most questions will be of the form
  - complete this program
  - what does this program display

Lab Quiz

- There will be a quiz in lab next week, September 20
- A lab quiz is similar to a regular lab, but you must do it all by yourself
- You may use your notes, textbook and the web
- Lab quizzes count for 5% of your total grade, each quiz is about 1% of your course grade

GUI Methods

- Important methods for a GUI program include
  - main
    - Creates an instance of the class
  - constructor (same name as class)
    - Initializes the GUI
    - Only called once
  - actionPerformed
    - Does what the program is supposed to do
    - Called whenever something is clicked
  - paint *(optional)*
    - Displays graphics
    - Called many times

Methods in any Order

```
public class MyProg {
        public static void main(String[] bird) {
                // main – program starts here
        }
        public MyProg() {
                // constructor – called once
        }
        public void actionPerformed(…) {
                // called for every click
        }
        public void paint(java.awt.Graphics canvas) {
                // called to repaint window
        }
}
```

Methods in any Order

```
public class MyProg {
        public void actionPerformed(…) {
                // called for every click
        }
        public static void main(String[] bird) {
                // main – program starts here
        }
        public void paint(java.awt.Graphics canvas) {
                // called to repaint window
        }
        public MyProg() {
                // constructor – called once
        }
}
```

Methods in any Order

```
public class MyProg {
        public MyProg() {
                // constructor – called once
        }
        public void actionPerformed(…) {
                // called for every click
        }
        public void paint(java.awt.Graphics canvas) {
                // called to repaint window
        }
        public static void main(String[] bird) {
                // main – program starts here
        }
}
```

What is displayed?

```
public class Colors {
   static void aqua() {
      System.out.println("Blue");
   }
  static void sun(){
      System.out.println("Yellow");
   }
   public static void main(String[] args) {
      rouge();
      System.out.println("Gray");
      sun();
   }
 static void rouge() {
      aqua();
      System.out.println("Scarlet");
   }
}
```

A. Blue
   Yellow
   Gray
   Scarlet

B. Scarlet
   Gray
   Yellow

C. Blue
   Scarlet
   Gray
   Yellow

D. Gray
   Yellow
   Scarlet

E. none of the above

paint Method

- The Java system calls the paint method when it wants to display the GUI

```
public void paint(java.awt.Graphics peacock ) {
      peacock.setColor(java.awt.Color.BLUE);
      peacock.fillRect( 10, 20, 100, 150 );
}
```

- The paint method may be called many times, particularly when you resize the window

setColor

- The setColor method of java.awt.Graphics objects sets the color that will be used when drawing shapes

```
bird.setColor(java.awt.Color.BLUE);
```

- where **`bird`** is an object of the type java.awt.Graphics
- Once you set the color, it stays that color

Graphics Coordinates

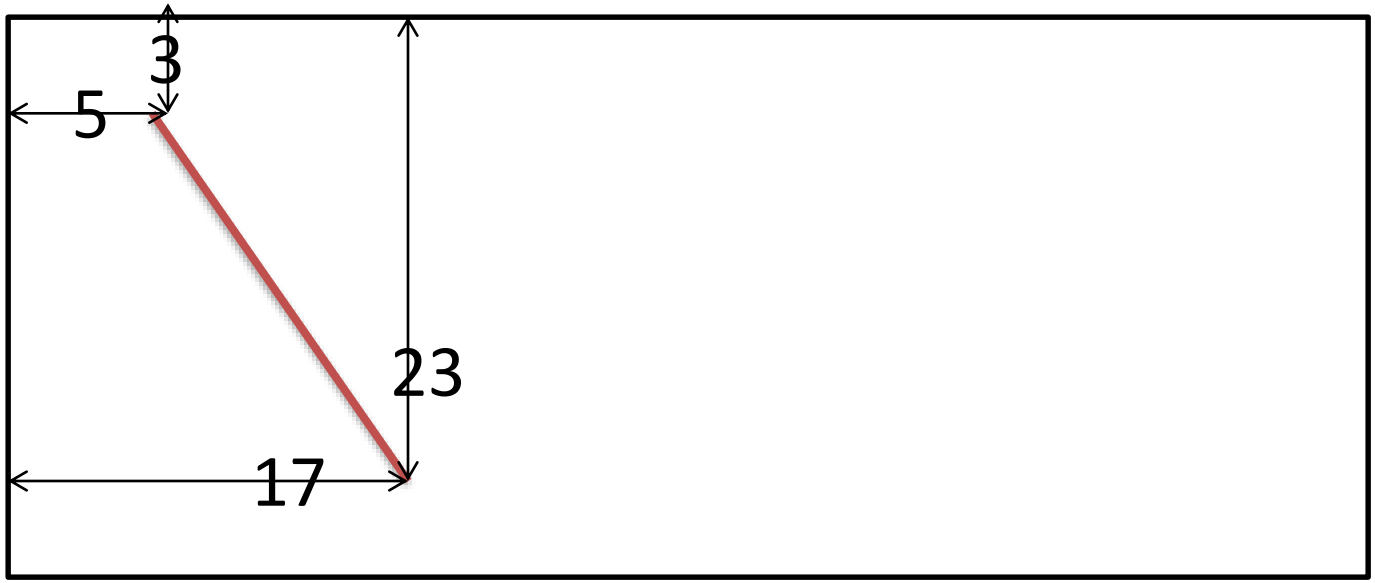
- The screen has a coordinate system with the origin in the upper left corner
- Coordinates are given in pixels (Picture Elements)
- An X coordinate specifies the distance from the left edge
- A Y coordinate specified the distance from the top edge
- The screen size depends on the device

drawLine

- The drawLine method of java.awt.Graphics objects draws a line from x_1,y_1 to x_2,y_2

```
bird.drawLine(5, 3, 17, 23);
```

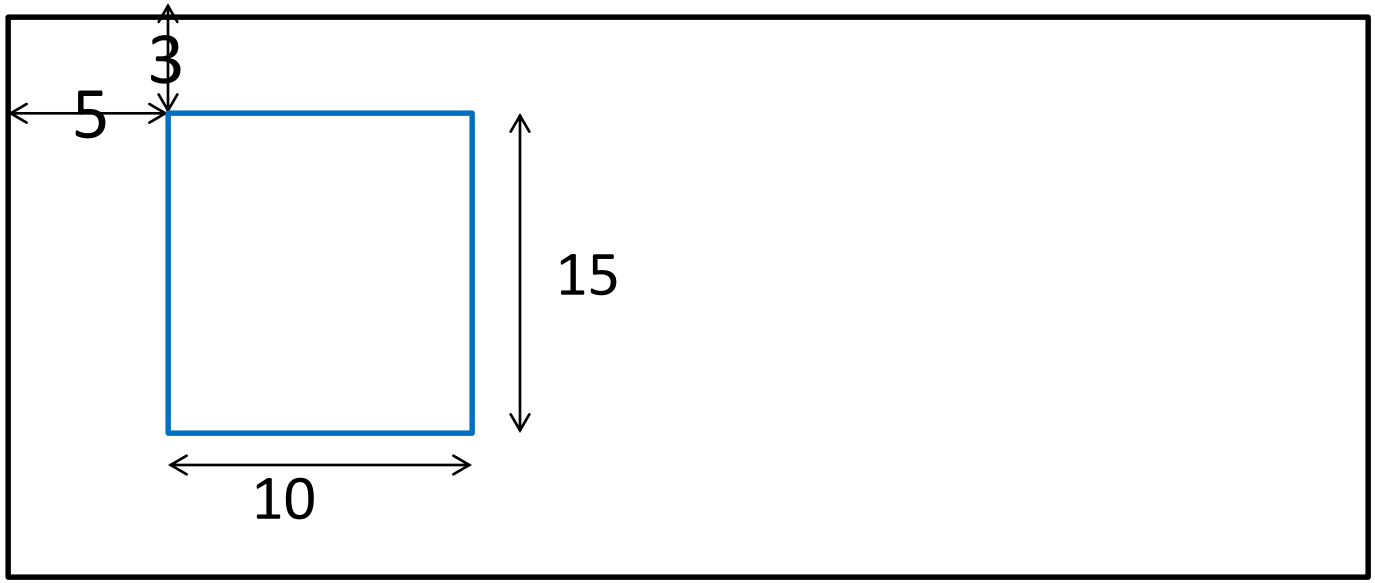
- The line is drawn using the current color



drawRect

- The drawRect method of java.awt.Graphics objects draws a rectangle of width and height whose upper left corner is x,y

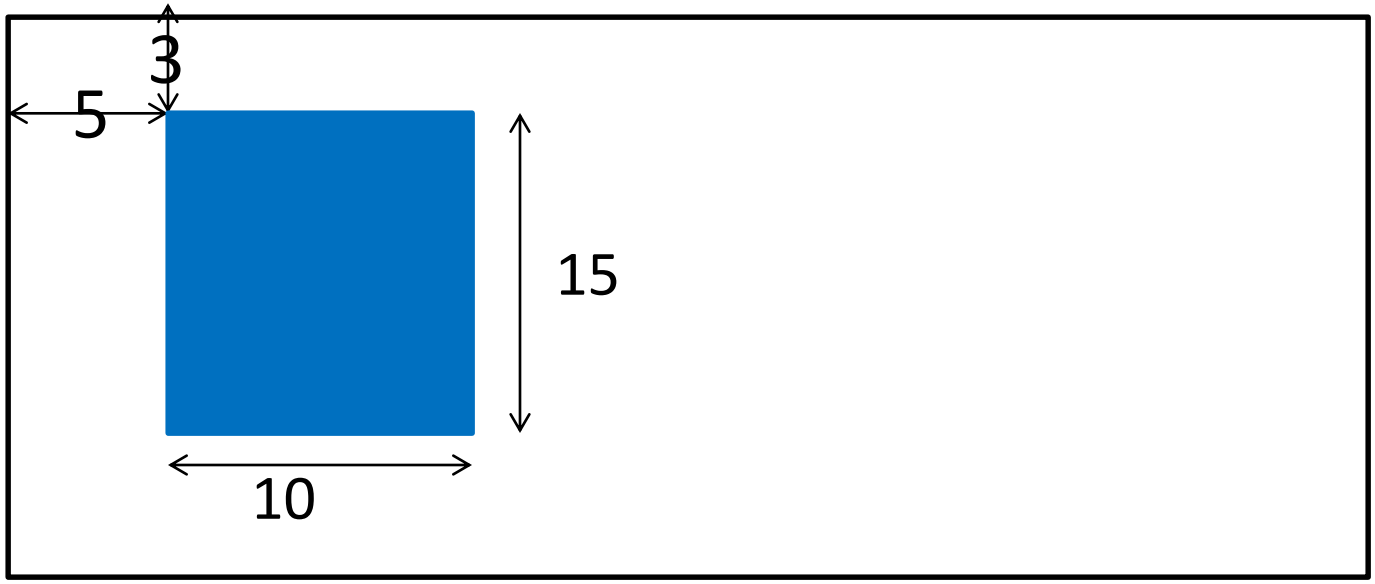
```
bird.drawRect( x, y, width, height);
bird.drawRect( 5, 3, 10, 15 );
```



fillRect

- The fillRect method of java.awt.Graphics objects draws a rectangle of width and height whose upper left corner is x,y and fills it with the current color

```
bird.fillRect( 5, 3, 10, 15 );
```



Outside and Overlapping

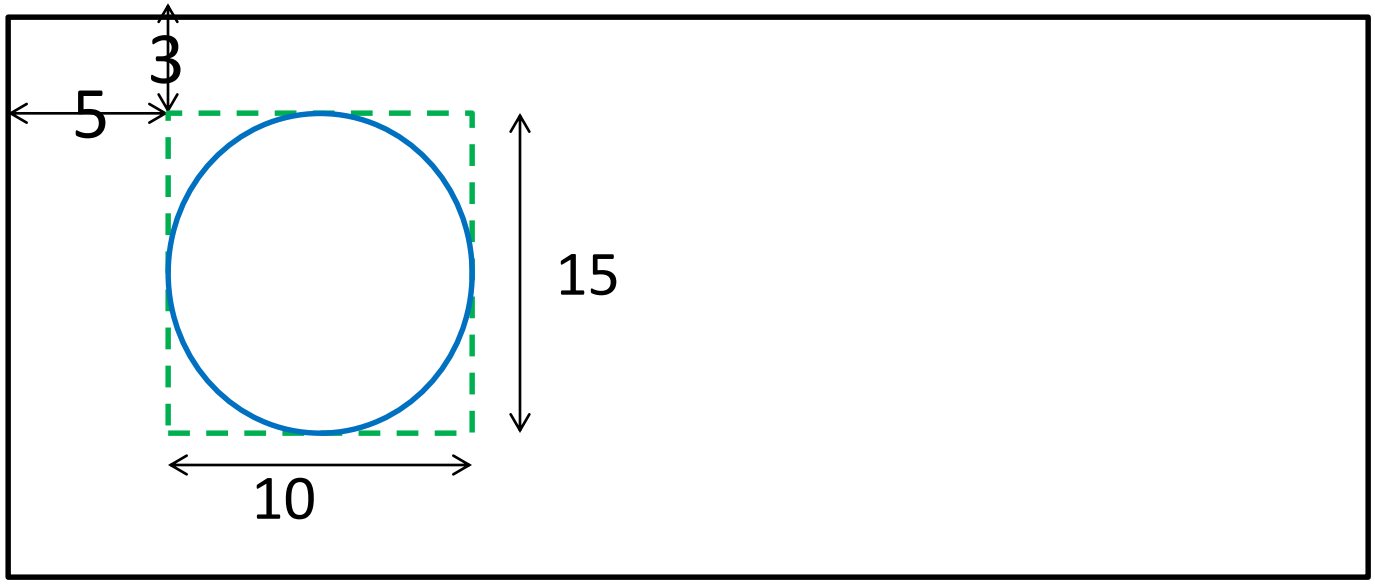
- It is permissible to have a graphics object (such as a rectangle) extend outside the visible window
- The portion outside the window will not be visible
- An object can overlap one another

```
bird.fillRect( 5, 3, 10, 15 );
bird.setColor( Color.GREEN );
bird.fillRect( 8, 15, 10, 15 );
```

drawOval

- The drawOval method of java.awt.Graphics objects draws a circle or oval to fit in a box of width and height whose upper left is x,y

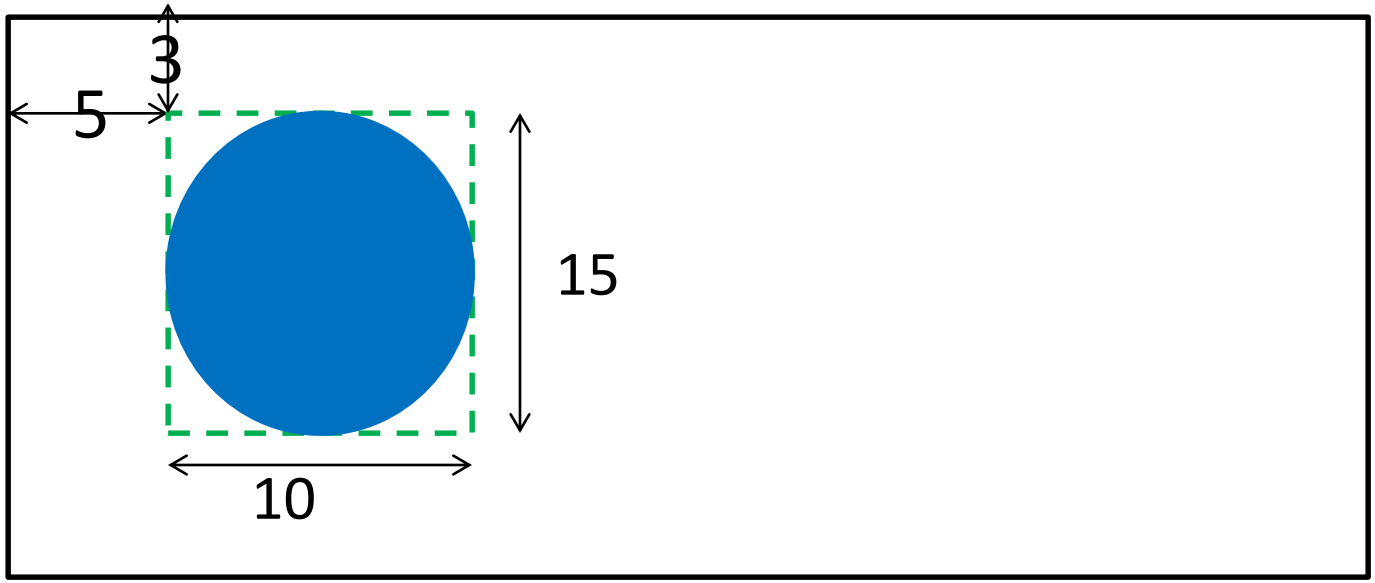
```
bird.drawOval( x, y, width, height );
bird.drawOval( 5, 3, 10, 15 );
```



fillOval

- fillOval is just like drawOval, but it colors in the circle with the current color

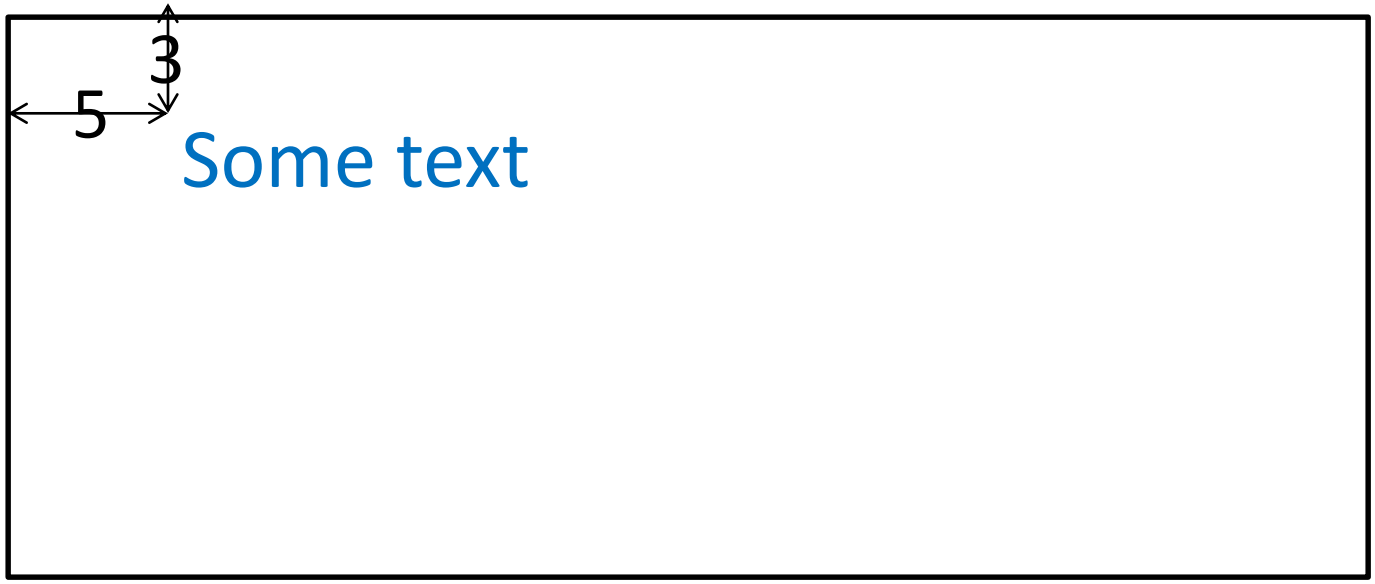
```
bird.fillOval( 5, 3, 10, 15 );
```



drawString

- The drawString method of java.awt.Graphics objects writes the text of a String starting at the specified x,y location

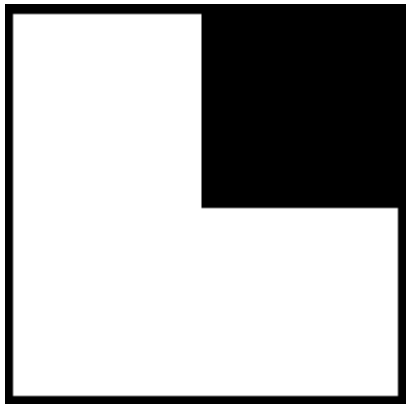
```
bird.drawString( String, x, y);
bird.drawString("Some text", 5, 3 );
```



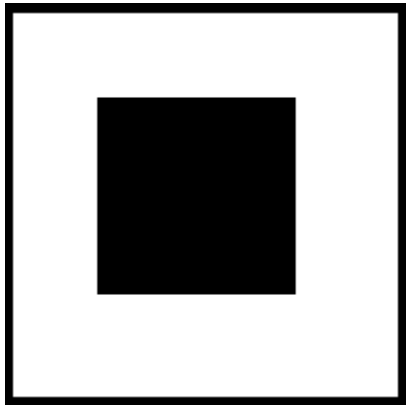
What is the output?

```
public void paint(java.awt.Graphics  parrot) {
        parrot.setColor(java.awt.Color.BLACK);
        parrot.drawRect( 0, 0, 100, 100);
        parrot.fillRect( 0, 50, 50, 50 );
}
```

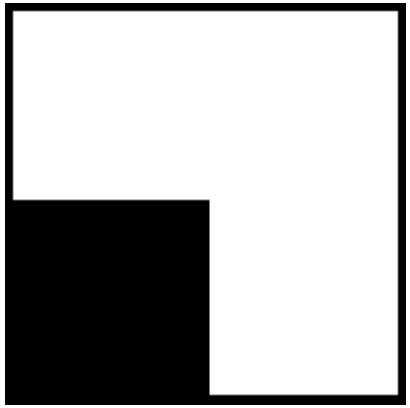
A.



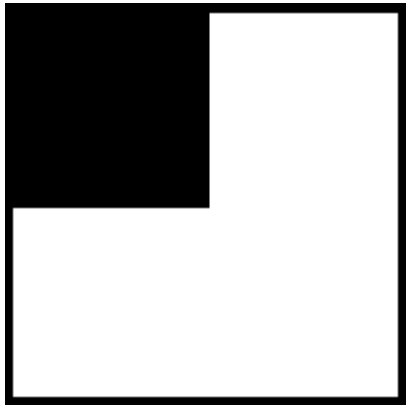
B.



C.



D.



Calling Methods

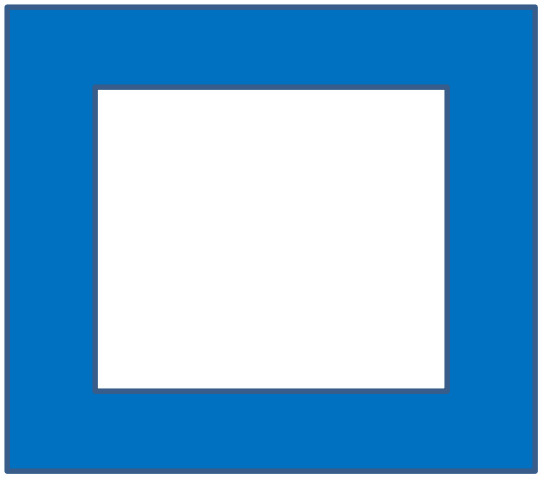
- The graphics programs tend to call many methods
- Most graphics methods do not return a value
- They are used on a line by themselves
- Methods that return a value are often used in an equation

With the students around you

Complete this method to draw a 150 by 150 pixel image of the nautical flag for the letter P

```
public void paint(Graphics flag) {
        // Your code goes here
}
```

P

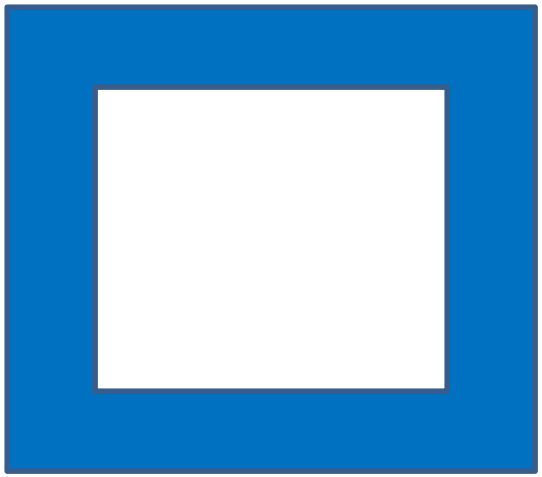


Possible Solution

Complete this method to draw a 150 by 150 pixel image of the nautical flag for the letter P

```
public void paint(Graphics flag) {
   flag.setColor(java.awt.Color.BLUE);
   flag.drawRect(0, 0, 150, 150);
   flag.setColor(java.awt.Color.WHITE);
   flag.drawRect(30, 30, 90, 90);
}
```

P



Methods with or without return value

- Value returning method

```
int blue = getBlue( x, y );
```

- void method that does not return a value

```
setBlue( x, y, blue );
```

Calling Methods of Classes

- Static methods are called on a class

Classname.method( )

- Math.cos(dog) is a static method

We have been using methods

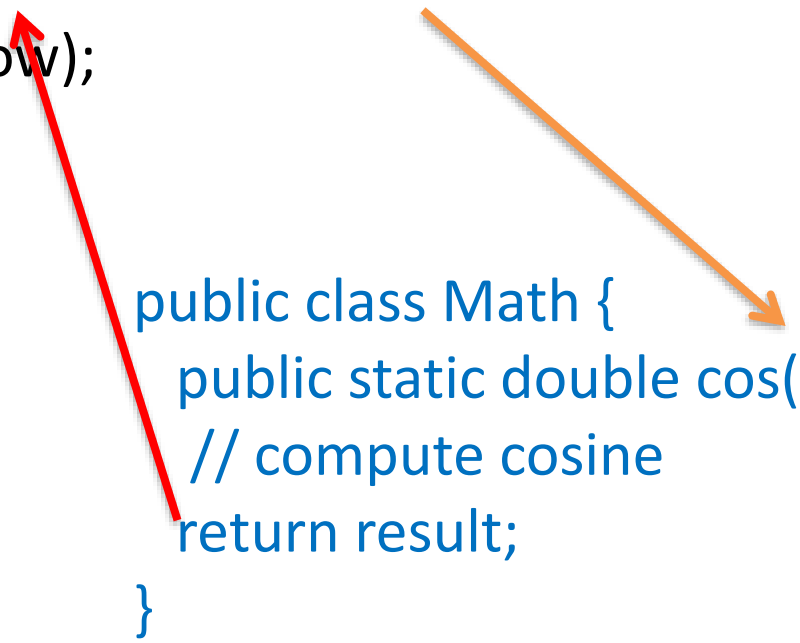
- Homework and lab assignments have used methods

```
cow = 2.0 * Math.cos( theta );
```

- During the execution of your program, it executes the cosine method which returns the result back to your program

Transfer of Execution

```
public class Demo {
  public static void main( String[ ] stuff) {
  double theta = 1.047;
  double cow = 2.0*Math.cos(theta);
  System.out.println(cow);
  }
}
```



Calling Methods of Objects

- Non-static methods are called on objects, not classes

```
public void paint(java.awt.Graphics   cow ) {
        cow.setColor(java.awt.Color.PURPLE);
        cow.fillOval( 10, 25, 7, 7);
}
```

Class Variables

```
public class Critter {
    int skink = 11;
    int doit( int toad ){
        int lizard = skink + toad;
        return lizard;
    }
}
```

- **skink** is an instance variable
- **toad** is a parameter variable
- **lizard** is a local variable

Instance Variables

- An instance variable (a.k.a. field variable, class variable or attribute) hold long term data
- Each object has its own copy of the instance variables
- Objects hold the data of a logical entity
- Instance variables can be used in all methods

Local Variables

- Local variables are defined in a method
- Local variables can only be used in the method where they are defined
- The values of local variables disappear when the program returns

Parameter Variables

- Parameter variables have an initial value from the calling program
- Parameters are the input to a method
- Parameter variables can only be used in that method
- Changing simple parameter variables does not change the calling program's variables

Making a GUI

1. Create a class that extends JFrame and implements java.awt.event.ActionListener
2. Create component objects
3. Specify the layout manager
4. Add the component objects to the container
5. Add an action listener to the components that respond to user input
6. All the program to be visible and stop
7. Create an actionPerformed method

Which is a local variable?

```
int  aardvark;
public class Click {
        int  bunny;
        public void aMethod( int cow ) {
                int dog;
        }
}
```

A. aardvark
B. bunny
C. cow
D. dog
E. aMethod

Usual Java GUI

```
public class MyProgram extends javax.swing.JFrame
    implements java.awt.event.ActionListener {
        // declare GUI components here
  public MyProgram() {
        // setup GUI here
  }
  public void actionPerformed(
                java.awt.event.ActionEvent  dog){
        // program here
  }
  public static void main(String[] cat) {
      MyProgram fish = new MyProgram();
  }
}
```

GUI Display

- GUI programs usually set the text of a JLabel

```
// in the class before any methods
JLabel hawk = new JLabel();
// in the actionPerformed method
double dog = Math.sin(cat);
hawk.setText("The answer is " + dog);
```

If a JFrame GUI program calls System.out.println, the program will

A. compiler error
B. run time error
C. display the data in the usual place for applications
D. do nothing

GUI Text Input

- A GUI usually gets text input from a JTextField

```
                    //  in the class before any methods
JTextField  kitten = new JTextField();
                    //  in the actionPerformed method
String bear;
bear = kitten.getText();
```

String and Numbers

- A String can contain any character on the keyboard, including the numbers

```
String myGrade = "4.0";
double myScore = 4.0;
```

- You cannot do arithmetic with Strings
- A string containing numerical characters must be converted to a double or an int for use in any calculations

Converting Strings to Numbers

- A String that contains numerical characters can be converted using

```
String deuce = "2.25";
double cow = Double.parseDouble( deuce );
String four = "4";
int goat = Integer.parseInt( four );
```

- These methods will throw an exception if the string does not contain a number

Converting Numbers to Strings

- There are several methods to convert numbers to Strings
- The easy way is to concatenate a number with a string

```
double dog = 47.5;
String lizard;
lizard = "The answer is " + dog;
```

Complete this program to read a number and display the square root

```
import javax.swing.*;
public class MyProgram extends JFrame
                              implements java.awt.event.ActionListener {
        JTextField crow   = new JTextField();
        JLabel       raven = new JLabel("answer here");
  public MyProgram() { /* Usual GUI initialization */ }
  public void actionPerformed(java.awt.event.ActionEvent  dog){
        // Complete this method
  }
```

Possible Solution

```
public void actionPerformed(java.awt.event.ActionEvent  dog){
        String rat         = crow.getText();
        double mouse = Double.parseDouble( rat );
        double shrew  = Math.sqrt( mouse );
        raven.setText("answer is " + shrew );
 }
```

Big Computer Science Concepts

- Inheritance
- Methods
- Constructor methods
- Creating objects
- Types of variables
- Strings are different from numbers

First Exam

- The first exam in COMP163 will be in lecture on Monday, September 17
- The exam will cover everything since the beginning of the semester
- You are allowed one 8½ by 11 inch page of notes
- Most questions will be of the form
  - complete this program
  - what does this program display

Lab Quiz

- There will be a quiz in lab next week, September 20
- A lab quiz is similar to a regular lab, but you must do it all by yourself
- You may use your notes, textbook and the web
- Lab quizzes count for 5% of your total grade, each quiz is about 1% of your course grade