- Previous Lecture:
  - Selection statement
  - Reading input using `Scanner` class
  - Iteration with `while` loop
- Today's Lecture:
  - Methods (functions)
  - Intro to objects and classes
- Reading:

Method

```
/* = a random integer in [lo..hi]
 */
public static int randInt(int lo, int hi) {
  return (int) (Math.random()*(hi-lo+1) + lo);
}
```


Method

A method is a named, parameterized group of statements

modifiers return-type method-name `(` *parameter-list* `) {`
    statement-list
`}`

return-type `void` means nothing is returned from the method

There must be a `return` statement, unless return-type is `void`

Method

A method is a named, parameterized group of statements

modifiers return-type method-name `(` *parameter-list* `) {`
    statement-list
`}`

parameter-list : type-name pairs separated by commas

```
int randInt(int lo, int hi)
```

A parameter is a variable that is declared in the method

Calling a `static` method …

… that is in a different class:

classname.*methodname*(…)

Examples: `Math.random()`
`Math.pow(2.5,2)`

… that is in the same class:

`methodname``(…)`

Our class `Prelim2Q1a` has a `static` method `randInt`, so an example method call within the class can be

`randInt(3,8)`

`Math` class

- A collection of common mathematical functions and constants
- `static` methods and constants
  - Belongs to the class
  - An object is not needed to access `static` members of a class

```
import javax.swing.*;

public class MakeFrame {
  public static void main(String[] args)
  {
    JFrame f= new JFrame();
    f.show();
    f.setSize(600,200);
    int w= f.getWidth();
    System.out.println("Width is " + w);
  }
}
```

Pre-defined class `JFrame`

- Deals with windows (frames) on the monitor
- All the predefined classes are collectively called the Java API
- Classes are grouped into packages. E.g., java.io, java.net, javax.swing
- Use the `import` statement:

```
import javax.swing.*;
```

- To find out what the classes do, read the API specifications:

  http://java.sun.com/j2se/1.5.0/docs/api

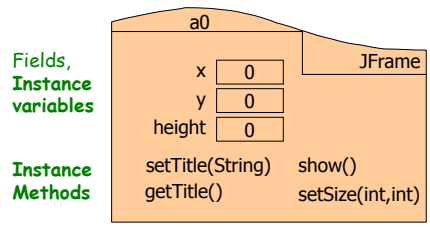
Object & Class—an analogy

- *Object*: a folder that stores information (data and instructions)
- *Class*: a drawer in a filing cabinet that holds folders of the same type

What is in an object?

(What is in a folder?)

- Fields to store data
- Instructions for dealing with the object



Creating an object

The expression

```
new JFrame()
```

- Creates a `JFrame` object (folder) and gives it a reference name
- Calls method `JFrame()` to set initial values for the object
- Yields the reference of the object

Reference variable

- Use a reference variable to hold on to an object:

```
JFrame f= new JFrame();
```

Use the class name as a type