Road Map

• Method Overloading
• Scope
• Pass-by-value
• Style: methods

• Reading:
  – Liang: chapter 5: 5.5. - 5.8
review

• When is the return statement required?
• What do the following method headers tell us?

```java
public static int max (int a, int b)
public static void nPrintln (String s, int n)
public static void main (String args[])
public static int max (int a, n)
public static String pickLine ()
```
Given the following method headers:

public static int max (int x, int y)
public static double min (double x, double y)

Which of the following method calls are legal?

max (1,2);
min (1,2);
max (1.0, 2.0);
min (1.0, 2.0);
Review

• What is the return type of Math.random()?
• What is the range of the result of Math.random()?
• What is the range of possible values of the following?
  \[ 1 + (\text{int})(20 \times \text{Math.random()}) \]
6.15 Method Overloading

• Method overloading
  – Several methods of the same name
  – Different parameter set for each method
  • Number of parameters
  • Parameter types
  – The Java compiler determines which method to use based on the parameters.
  – Can also be used in conjunction with argument coercion.
  • The combination can lead to ambiguous invocation which is an error
Understanding Scope
Scope

• Determines *where* the variable can be referenced in a program.
• By understanding scope, you can make your programs:
  – more modular
  – easier to debug
6.9 Scope of Declarations

- **Scope**
  - Portion of the program that can reference an entity by its name
  - Basic scope rules
    - Scope of a parameter declaration
      - The entire method
    - Scope of a local-variable declaration
      - From where it is declared until the end of the block it is declared in
    - Scope of a local-variable declaration that appears in the initialization section of a for statement’s header
      - The entire code block of the for loop
pass by value

• In Java when a parameter is passed to a method and the variable is modified by that method, the value does not change upon return to the calling method
  – There are exceptions to this which we will discuss later
• This is not true in all programming languages.
Document your methods

• An acceptable way is to precede the declaration with the following:

```java
/**
 * Returns the tax due based on the parameters.
 * Both parameters must be non-negative
 * @param income dollar amount of sale
 * @param taxRate tax rate for sale
 * @return the tax due
 */
public static double calculateTax (double income, double taxRate)
{
    ...
    code....
} /* end method calculateTax */
```
Good Programming Habits

• Pick a method name that strongly, completely and clearly describes what the function does or returns
  – E.g. verb-plus-object or description of returned value (getters, setters, predicates are object orientated naming conventions)

• Make sure your actual parameters (i.e. the actual data you pass to a method) matches the data type the method expects.

• Make sure you use all the parameters in your method; if not, lose the ones you don’t use.

• Document any assumptions about your parameters!
More Good Programming Habits

• Limit the number of a method’s parameters to less than seven (preferably a lot less...)
  – Studies have shown that people can’t keep track of more than seven pieces of info at once
  – (e.g. 7 digit phone numbers for a reason)

• If a method is to return a value, make sure it returns something under all possible circumstances.
Superman program

- Superman needs a program to print S’s of various size. You should have one method that prints the horizontal lines and one method that prints the vertical lines. Write the whole program.
  - Vertical lines can be in the first or last column and are width/2 - 1 characters tall.
  - For example, an S of size 6 would look like:

```
SSSSSSSS
SSSSSS
SSSSSS
```

```