

## CREATING MYLIB AND TESTING YOUR RANDOMNUMBER() METHOD

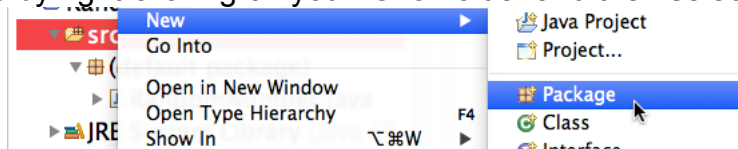
- Your code should start out looking something like this:

```

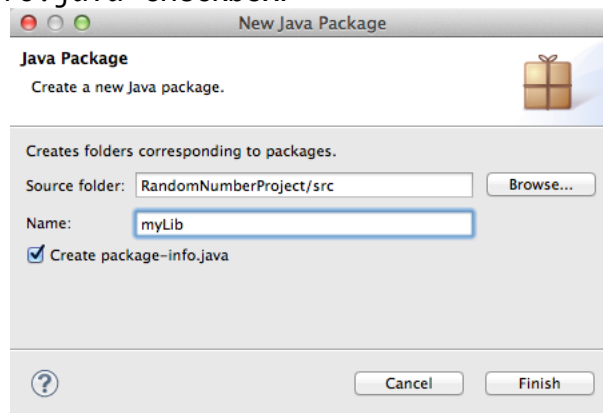
01 public class RandomNumber {
02     public static void main(String[] args) {
03         int i=0;
04         while (i<10) {
05             int r=randomNumber(1,10);
06             System.out.println(r);
07             i++;
08         }
09     }
10     public static int randomNumber(int from, int to) {
11         int range=to-from+1;
12         int rand=(int)(Math.random()*range);
13         int result=rand+from;
14         return result;
15     }
16 }

```

- You will now be creating a "Library" of methods you can use for the rest of the class much like the Math library you have been using. You begin by creating a new package called "myLib" in eclipse by right-clicking on your "src" folder and then selecting "new" and "package"



- In the dialog box that appears enter "myLib" in the "Name:" field and **be sure to check** the "Create package-info.java" checkbox!



- You will see the code below:

```

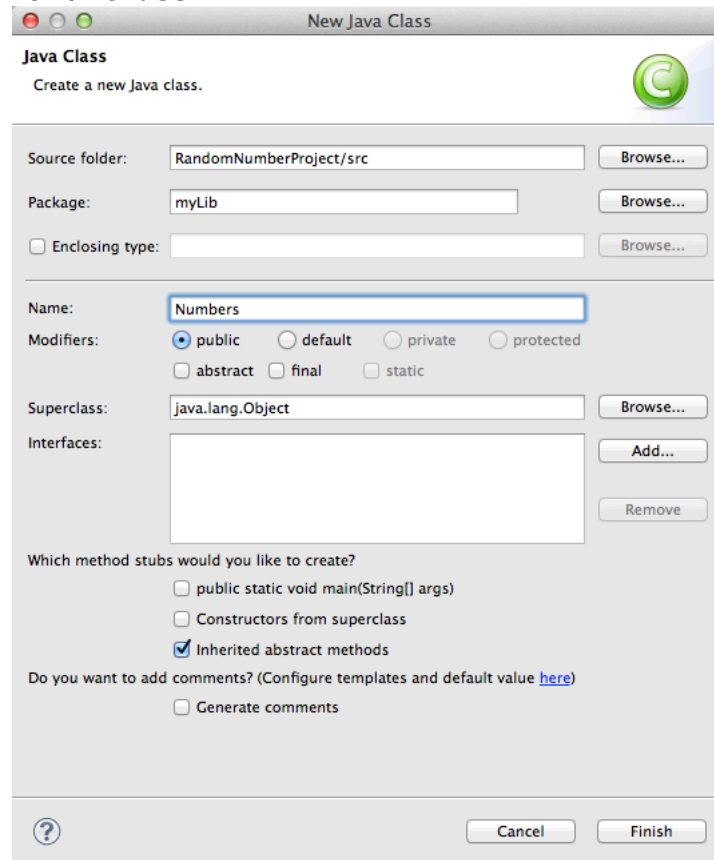
01 /**
02  *
03  */
04 /**
05  * @author hyperion
06  *
07  */
08 package myLib;

```

You will need to make the following changes:

- Delete the comments on lines 01 to 03.
  - Change the text after the **@author** to include your name in Last, First (Period x) format.
  - Below the author, add the line:  
**@version** Last Modified Month, Day, Year.
  - Below the version line, fill in a description of this library indicating that this is your library of code for AP Computer Science A on a line with just a \*
- Next you will create a class to hold **your** version of the randomNumber() method.

- Create a new class in the "myLib" package called "Numbers" by right-clicking on the package and selecting "new" and "class"



- Select your randomNumber() method from your code window and cut it from your program:

```
public static int randomNumber(int from, int to) {
    int range=to-from+1;
    int rand=(int)(Math.random()*range);
    int result=rand+from;
    return result;
}
```

- Paste the method into the Numbers class:

```
package myLib;

public class Numbers {
    public static int randomNumber(int from, int to) {
        int range=to-from+1;
        int rand=(int)(Math.random()*range);
        int result=rand+from;
        return result;
    }
}
```

- Now you will need to fix your original main() method by modifying the method call that no longer works: `int r=randomNumber(1,10);` so it reads:

```
int r=myLib.Numbers.randomNumber(1,10);
```

- You can now run your program as before.

- Next we will add JavaDoc comments for your method. You can see a sample JavaDoc comment that is far more comprehensive than you will need to write at this stage by hovering over the `random()` method in your code:

```
double java.lang.Math.random()

random
public static double random()

Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0. Returned values are chosen pseudorandomly with (approximately) uniform distribution from that range.

When this method is first called, it creates a single new pseudorandom-number generator, exactly as if by the expression

    new java.util.Random

This new pseudorandom-number generator is used thereafter for all calls to this method and is used nowhere else.

This method is properly synchronized to allow correct use by more than one thread. However, if many threads need to generate pseudorandom numbers at a great rate, it may reduce contention for each thread to have its own pseudorandom-number generator.

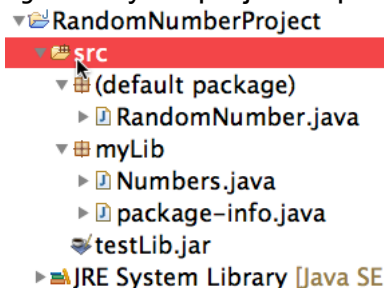
Returns:
    a pseudorandom double greater than or equal to 0.0 and less than 1.0.

See Also:
    Random.nextDouble\(\)
```

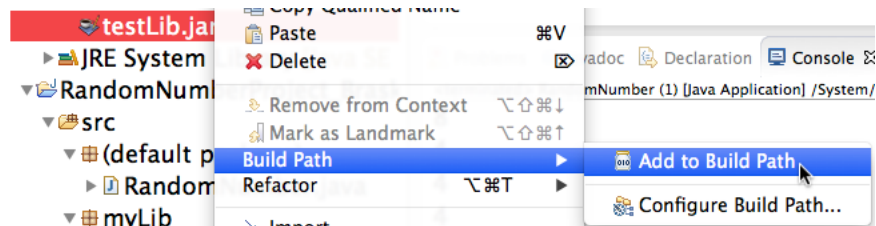
- You start a JavaDoc comment by inserting a new blank line before your method declaration:  
`public static int randomNumber(int from, int to) {`
- You then type a multi-line comment but with two asterisks after the forward slash: `/**` and hit return or enter.
- You should see the following lines of code appear:

```
/**
 *
 * @param from
 * @param to
 * @return
 */
```

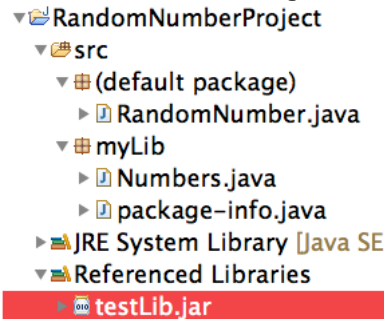
- On the first line with nothing but an asterisk you should write your description of what this method does.
- At the end of the line beginning: `* @param from`, write a description of what this parameter represents.
- Do the same thing for the parameter `to`.
- Finally, write a description of the return value of this method after the `@return`.
- Test your JavaDoc comment by hovering over the method call to `randomNumber()` in your code.
- Next we need to import the test suite for this project. Open the "Libraries" folder inside the "AP\_CompSci" folder and select the file "testLib.jar"
- Copy it using a keyboard shortcut or right clicking the file and then selecting copy.
- Switch back to Eclipse and select the "src" folder, then paste using a keyboard shortcut or right-clicking and selecting paste.
- You should now see "testLib.jar" in your project explorer:



- To gain access to the code in this library, you right-click "testLib.jar" and select "Build Path" then "Add to Build Path"



- Your project explorer should now show "testLib.jar" in your "Referenced Libraries"



- Now modify your main() method so it calls the Test Suit by replacing your original code with this code:

```

01 import testLib.Test;
02
03 public class RandomNumber {
04     public static void main(String[] args) {
05         Test.testRandomNumber(1000, null);
06     }
07 }
  
```

- If you are curious, hover over the testRandomNumber() method to read JavaDoc.
- When you run this code, you should see something like the console output below:

```

Testing myLib.Numbers.random(from,to) using 1000
trials.
e=Expected, a=Actual
From To    eMin  aMin  eMax  aMax  Pass/Fail
  1  10     1     1    10    10    Passed
-10 -1     -10   -10   -1    -1    Passed
 10  5       5     7    10    10    Failed
  1  1       1     1     1     1    Passed
-1  -1     -10   -8    -1    -1    Failed
  1  -1     -1     1     1     1    Failed
  0  0       0     0     0     0    Passed
Score: 4/7
  
```

- You are now ready to have your results scored by the teacher!