**Lecture/Discussion**

1. What is a method?
   1. Purpose
      1. Run a set of instructions.
      2. Used to compartmentalize code.
      3. Optionally receives an input and returns an output.
      4. Usually optimized for generic case use.
   2. Method Declaration
      1. **public static int** add(**int** x, **int** y) {}
      2. Code for method goes within curly braces, {}
   3. Method Signature
      1. **public static int** add(**int** x, **int** y) {}
      2. Comprised up of method name and parameters.
   4. Parameters
      1. The expected inputs of the method.
      2. Data type must be specified.
   5. Arguments
      1. The passed inputs. The actual values used when calling the method.
      2. Number and type of arguments must match that of the parameters for the method being called.
   6. Return Type
      1. The output of the method.
      2. Data type must be declared in the method declaration beforehand.
      3. The “return” keyword is used to get/return the output of the method. This keyword must exist in the body of the method.
         1. Jumps out of the method at this keyword.
      4. Some return types: int, String, double, int[], Object, etc.
      5. “void” means no return type. The “return” keyword does not have to be used.
2. Usage
   1. Methods must be called from another part of the program, like so:
      1. add(3, 5); *// 8*
   2. Method that return values can be assigned to variables.
      1. **int a** = add(3, 5); *// a now has 8*System.out.println(a); *// 8*

**In Class Examples**

Example 1: Add Method

*/\*\*  
 \* Adds two integers together  
 \** ***@param x*** *The first int  
 \** ***@param y*** *The second int  
 \** ***@return*** *The sum  
 \*/***public static int** add(**int** x, **int** y)  
{  
 *// Store the sum of x and y into a variable* **int** sum = x + y;  
 *// Return the sum, which is the output* **return** sum;  
}

Example 2: Lowest Score Method

*/\*\*  
 \* Gets the lowest int from an array of ints  
 \** ***@param n*** *The array of ints to look through  
 \** ***@return*** *The lowest int in n  
 \*/***public static int** getLowest(**int**[] n)  
{  
 *// In the beginning, the first number in n is automatically the lowest,  
 // Since it is the only number that has been looked at during this point* **int** lowest = n[0];  
  
 *// Loop i from 1 to the total number of ints  
 // The loop control variable, i, represents  
 // the current index of the array of numbers, n* **for**(**int** i = 1; i < n.**length**; i++)  
 {  
 **int** currentNumber = n[i];  
 *// If the currentNumber is lower than the lowest score,* **if**(currentNumber < lowest)  
 {  
 *// Then by defenition, the currentNumber is the lowest score* lowest = currentNumber;  
 }  
 }  
 *// Return the lowest int* **return** lowest;  
}

**Practice**

1. Which of the following is the valid method declaration for the given method?
2. public static int printText(String text)
3. public static int add(x, y)
4. public static printText(String text)
5. public static double multiply(double x, double y)
6. Describe what the arguments and parameters of a method are.
7. Write a method called subtract, that subtracts two doubles and returns the difference. For example, subtract(7.7, 3.1) should return 4.5. Be sure to include comments in your solution!
   1. What are the required parameters of the method?
   2. What is its return type?