CS 152 Computer Programming Fundamentals
Project 4: Welcome to Methods

Brooke Chenoweth

Spring 2025

1 Assignment Description

I have provided you with some starter code in a file named `MethodsPractice.java`. Place this file in your project in IntelliJ and you use it as a starting point. Using this code, complete the methods as written in the descriptions provided below. There are eight methods which must be written, five of which must be written completely from scratch.

The program will test your methods and report how many methods produced correct results. Submitting code that compiles, and passes tests for 8/8 methods is expected.

1.1 `minOfThree`

The `minOfThree` method takes three integer arguments and returns the smallest of them.

A skeleton of this method complete with a Javadoc comment is already in the java file, so all you need to do is replace the method body.

1.2 `reverseString`

The `reverseString` method takes a string and returns a new string with the characters in reverse order.

A skeleton of this method complete with a Javadoc comment is already in the java file, so all you need to do is replace the method body.

1.3 `isLetterOrDigit`

The `isLetterOrDigit` method takes a character and returns true if the given character is a letter or a digit and false otherwise.

You have to implement this method from scratch. Do not forget the public static modifiers. Make sure you include a Javadoc comment before the method.

1.4 `getLettersAndDigits`

The `getLettersAndDigits` method takes a string and returns a new string containing only the letters and digits from the argument, removing whitespace, punctuation, and any other symbols/characters that may be present in the original string.

You have to implement this method from scratch. Do not forget the public static modifiers. Make sure you include a Javadoc comment before the method.

1.5 `isPalindromePhrase`

The `isPalindromePhrase` method takes a string and returns true if the string is a palindrome phrase, false if not.

A *palindrome* is a sequence of symbols that reads the same backwards as forwards. For identifying a palindrome phrase in this method, we will ignore whitespace, punctuation, and capitalization, and consider only the order of the letters and numbers within the string.

You have to implement this method from scratch. Do not forget the public static modifiers. Make sure you include a Javadoc comment before the method.

1.6 `averageOddNumbers`

The `averageOddNumbers` method takes five integers and returns the arithmetic mean of the odd values. Any even values are not included in the average. If none of the values are odd, returns -1000.

A skeleton of this method complete with a Javadoc comment is already in the java file, so all you need to do is replace the method body.

1.7 `getTotalBill`

The `getTotalBill` method takes two arguments, an int meal which is the cost of the meal and a double tip which represents the tip percentage one would add to the bill, and returns a double which equals how much should be paid. The value of meal must be greater than zero and the value of tip must be greater than or equal to 0 and less than or equal to 0.9 (no tips over 90%). If either number is invalid, return -1.

You have to implement this method from scratch. Do not forget the public static modifiers. Make sure you include a Javadoc comment before the method.

1.8 `solveQuadratic`

A *quadratic equation* can be written as

$$ax^2 + bx + c = 0$$

where x is an unknown number and $a \neq 0$, b, and c are known values. The solutions of this equation can be computed using the *quadratic formula*

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The `solveQuadratic` method takes three doubles for the three coefficients of a quadratic equation and returns the larger of the two solutions computed with the quadratic formula. You may assume that we will only call this method with arguments that result in real results, so no need to worry about dividing by zero or taking the square root of a negative number.[1]

You have to implement this method from scratch. Do not forget the public static modifiers. Make sure you include a Javadoc comment before the method.

2 Turning in your assignment

Submit your `MethodsPractice.java` file to the Project 4 assignment in Canvas. Do not attach `.class` files or any other files.

3 Grading Rubric (total of 30 points)

[-5 points]:	File submitted to Canvas was not named `MethodsPractice.java`
[-5 points]:	The code did not compile or compiled with errors or warnings.
[6 points]:	The code adheres to the coding standard specified on the course website.
[24 points]:	Passes all tests (3 points per method) You will not receive full credit for a method if you merely hard code the test case output rather than correctly implementing the described behaviour. (We may run additional/different tests to verify this.)

[1] Feel free to check for those cases and print an error message and/or return some sort of error value if you like!