Lab 14

**Rules:**

1. You are asked to implement **Library.java** as described below.
2. Do not forget to take your work with you when you leave the lab by either copying your work files to your own USB flash disk, or by e-mailing them to yourself.

You are asked to implement the following **public static** methods and put them inside a class named “**Library**”. Here is the list of methods you need to implement and some description as to how they should work:

|  |  |
| --- | --- |
| int **max**(int a, int b, int c, int d) | Returns the maximum of the four arguments  E.g. max(3, 7, 1, 4)🡪7 |
| double **quadratic**(double a, double b, double c, double x) | Returns the value of the quadratic equation: ax2 + bx + c  E.g. quadratic(1.0, 2.2, 1.21, 0.1) 🡪1.44 |
| int **firstDigit**(int num) | Returns the leftmost digit of the argument. Assume the argument is positive.  E.g. firstDigit(435236)🡪4 |
| int **oddMinusEven**(int num) | Returns the sum of the odd valued digits minus the sum of the even valued digits of the positive integer parameter.  E.g. oddMinusEven(43598)🡪(3+5+9)-(4+8) = 5 |
| double **countChange**(int q, int d, int n, int p) | Returns the value of q quarters, d dimes, n nickels, and p cents into dollars.  E.g. countChange(10, 5, 1, 2) 🡪3.07 |
| String **parityCheck**(int num) | Returns :   1. “**Odd Parity**” if all the digits of the parameter are odd 2. “**Even Parity**” if all the digits of the parameter are even 3. “**Mixed Parity**” otherwise   E.g. parityCheck(123456)🡪Mixed Parity  parityCheck(13595)🡪Odd Parity  parityCheck(224628)🡪Even Parity |
| int **digitMatch**(int a, int b, int c) | Returns the number of positions where the digits of all three numbers match.  E.g. digitMatch(423, 463, 1413)🡪2 |
| int **removeOddDigits**(int num) | Returns the positive integer parameter with all odd digits removed. Assume that “num” will be <= 10^7  E.g. removeOddDigits(458721)🡪482 |

To test your class, we are giving you a driver code (**Test.java**) that tests each of the methods in **Library.java** and prints your lab score on the screen. You are advised to implement your own test code. When grading, we may use a different Test. Make sure that your code works under all circumstances

Lab Work Submission:

* You can continue to work on this lab after our lab class, on your own, at home.
* Submit your lab work via Blackboard on or before: **Wednesday, October 25, 2023, 11:59pm**.
* The only accepted submission method!
* Once you submit your assignment you will not be able to resubmit it!
* Make absolutely sure the Java files you want to submit are the Java files you want graded.
* You will not be able to submit your lab work under any circumstances once **Lab14** disappears at **12:00 a.m.** on **Thursday, October 26, 2023**.
* There will be **NO** exceptions to these rules!
* To submit your lab work, upload **Library.java** (**with .java extension**) you did for this lab to the **Lab14** assignment in the **Labs** tab in your Lab section’s presence in Blackboard.
* Then, make sure you click the **Submit** button to submit your lab work.
* This lab is worth **5 points**.