Computer Science Department

COMP2310 (Fall 2022/2023)

Assign # 1 _Due Date: Sat. 26/11/2022 by 10:00 pm (on Ritaj)_

Notes:

1. The assignment should be submitted by the due date and time *(Late Assignments will not be accepted for any reason) on Ritaj.*
2. The assignments are _individual_ effort and copying the assignment will be treated as a cheating attempt, which may lead to *FAILING* the course.

_____

Write a java program (***using eclipse***) that reads the name (only first name) and id number (integer) of a user and then does the following:

a- Checks whether the id number is balanced or not. Assume a balanced id number is one that has the sum of its even digits equal to the sum of its odd digits taking into account that the id number does not have any digit repeated consecutively. For example, the ids 123581, 718, 8350, 72326 and 40691 are all valid id numbers while the ids 234, 123316, and 7665 are not. Your program should use a method called ***isBalanced*** to check and return whether a given id number is balanced or not.
Your program should refuse all invalid id numbers with an appropriate message and should keep asking the user to reenter the id number until a valid id is entered or the user enters an id of *-1* to exit program.

b- After entering a valid id number, your program should display a menu asking the user what he/she wants to do next. The menu should have the following options:

1- Solve ***any*** quadratic equation by entering the value of ***x*** as well as the values of the coefficients (***a,b,c***) of the equation. Your program should have a method called ***solveEquation*** to solve the equation and return the result.
(e.g. if the user enters equation values a=3, b= -8, c=2, and x= 3 then the result should be 5). *Hint: A quadratic equation is one of the form:*
$$ax^2 + bx + c = 0$$

2- Find the root values of any given quadratic equation entered as in item 1 above (only coefficients entered and no ***x***) . Your program should use a function called ***findEquationRoots*** that takes the equation values and prints the roots. Make sure

you check for imaginary roots and print a message indicating if any exist. (hint: use equation $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$)

(e.g. if the user enters equation values a=1, b=5, and c = 6 then the printed roots should be x = -2 and x = -3).

3- Combine any two quadratic equations with the same **x** value(s) into one equation. Your program should have a method called ***combineEquations*** that receives the coefficient values for any two equations and prints the result of combining the two. (e.g. If the user enters the following values for the first equation a=2, b= -3 and c= -2 and for the second equation a=5, b=0, and c = -20 then the program should combine the two equations and print the resulting equation as follows: ***7x2 – 3x -22***)

4- Exit menu.

Your program should keep displaying the menu above and doing the requested user's tasks until the user selects to exit the menu (item 4 above).

The program should not exit after selecting to exit the menu, but should keep asking the user to continuously enter a new name and valid id number and select items from the menu until the user enters an id number of -1.

VERY IMPORTANT:

1. Your project folder should be compressed (.rar) and saved as ***ass1_youridnumber_yourLabsectionnumber.rar*** (for example if your student id number is 1212345 and your lab **section** is section 4 then the assignment project folder should be called ***ass1_1212345_s4.rar***). Turn in your assignment by ***replying to the course coordinator's message*** on Ritaj and attaching your code .rar file (***ass1_youridnumber_yourLabsection.rar***).

2. Your project folder should include at least two classes: class **Equation** that includes all the methods described above and class **CheckEquation** that has the main method for testing the program.

3. You must include your full name, student id number, and lecture + lab section numbers in a comment at the beginning of ***each*** of your class files above.

4. Late/Wrong assignments (even one minute late), .rar files that cannot be decompressed (wrong format), or assignments *not sent as a reply to the coordinator's message* will NOT be graded and will

receive a zero grade. NO EXCEPTIONS for ANY REASON.