DEPARTMENT OF INFORMATION TECHNOLOGY (TSING YI)

HD in T&N (IT114103) and HD in DSA (IT114116)

ITE3101 Introduction to Programming - Assignment

Due Date for Submission: 26 Nov 2019 (Tuesday) 4:30pm (through Moodle)

Notice to Students

1. This assignment should be written by individual student. All downloaded materials are not allowed.

2. Plagiarism will be treated seriously. All assignments that have been found involved wholly or partly in plagiarism (no matter these assignments are from the original authors or from the plagiarists) will score **ZERO** marks.

3. Your program must use Java JDK8 or above to develop.

4. Your program must be structured and well commented.

   The first few lines in the source file must be comments stating the name of the source file, student name, student ID, course name, course code, and brief description of your program.

```
/*********************************************************
    Name of Source File:   MathQuiz.java
    Student Name (ID):     Chan Tai Man (191414141)
    Course Name (Code):    HD in TN (IT114103)

    Program Description:
       This program is . . . . . .
       . . . . .
*********************************************************/
```

   Marks will be deducted if such comments are not included.

5. Write down your test cases and the reason(s) for conducting each test case. Test your program by using your test cases.

6. You are required to hand in a zip file (under your name and class – e.g. *TN1A_ChanTaiMan.zip*) containing the following 2 items via the assignment link in Moodle. Late submission may score ZERO mark.

   a. Your java file(s) for program codes, including detailed comments
   b. MS word document for your testing (including the screen dumps as evidence of testing)

7. Weight of this assignment is 20% of the module total assessment.

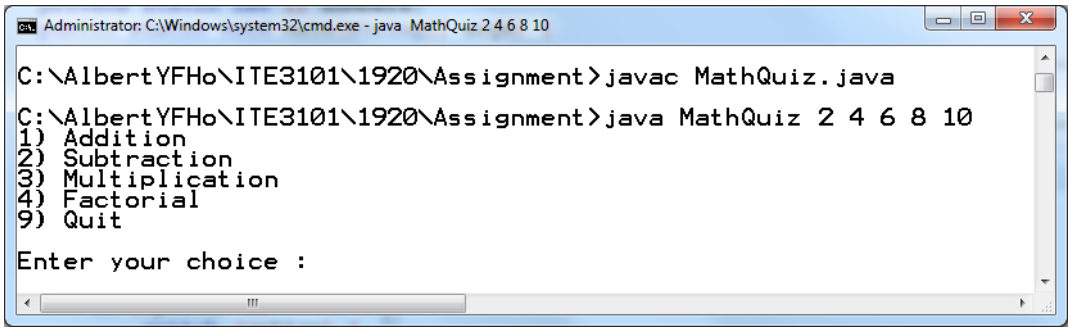
Mathematics Quiz

Problem Specification

You are asked to design and develop a Java Program that randomly generates Mathematics questions and computes the percentage of correct answers.

Step 1: Main Menu

When the program starts, a Main Menu will be shown as follow:

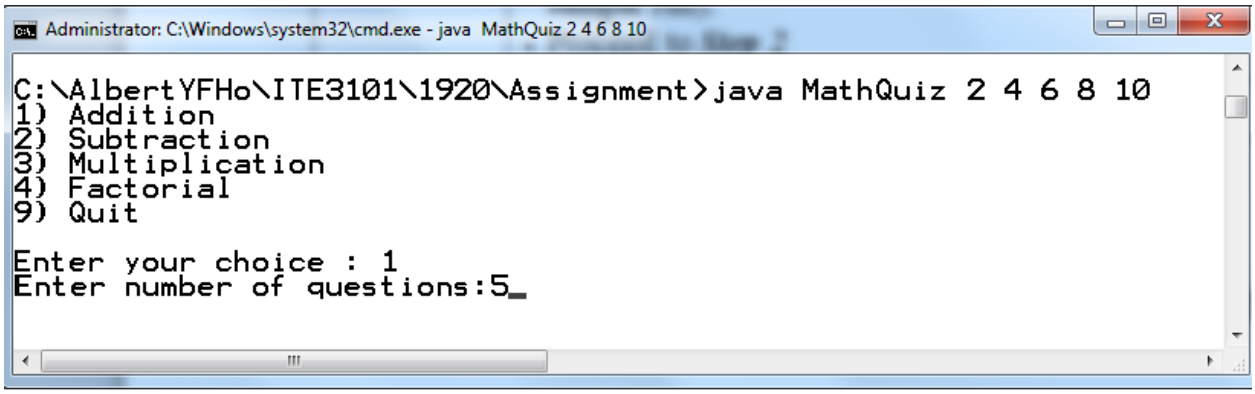


You have to handle the following user input:

User Input	***Action***
1	• Questions generated in ***Step 3*** will be questions on Addition of 2 numbers. The 2 numbers are randomly selected from the numbers input in the command line when the program runs (that is 2, 4, 6, 8, 10 in the above sample run). • Proceed to ***Step 2***
2	• Questions generated in ***Step 3*** will be questions on Subtraction of 2 numbers. The 2 numbers are randomly selected from the numbers input in the command line when the program runs (that is 2, 4, 6, 8, 10 in the above sample run). **An additional requirement is the first number should not be less than the second number under any circumstances**. • Proceed to ***Step 2***
3	• Questions generated in ***Step 3*** will be questions on Multiplication of 2 numbers. The 2 numbers are randomly selected from the numbers input in the command line when the program runs (that is 2, 4, 6, 8, 10 in the above sample run). • Proceed to ***Step 2***
4	• Questions generated in ***Step 3*** will be questions on Factorial of a number. The number is randomly selected from the numbers input in the command line when the program runs (that is 2, 4, 6, 8, 10 in the above sample run). • Proceed to ***Step 2***
9	• Proceed to ***Step 4***

Step 2: Specify the number of questions

After selecting the type of question successfully in ***Step 1***, we specify the number of questions as follow:

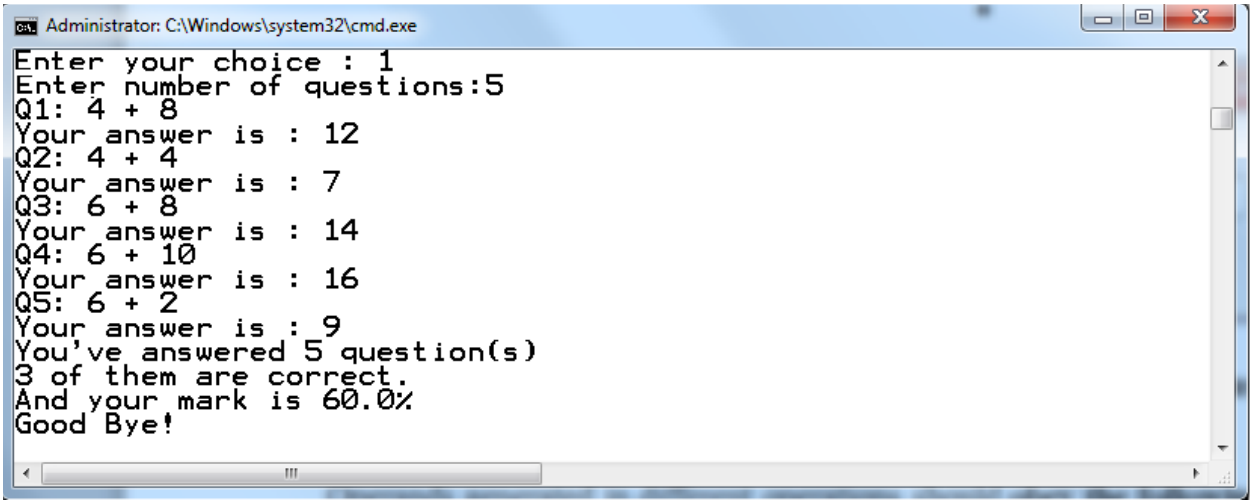


For example, if the user input is 5, then 5 *questions* of the chosen type (questions on addition in this example) will be asked in ***Step 3***.

Step 3: Generate Random Questions, let the user answer the questions and show the result

n questions will be generated, where ***n*** is specified in ***Step 2***. The math operator used is based on the input in ***Step 1***. For example: If we input 1 in ***Step 1*** (*i.e. Addition*) and input 5 in ***Step 2***, then ***Step 3*** will ask 5 questions on addition.

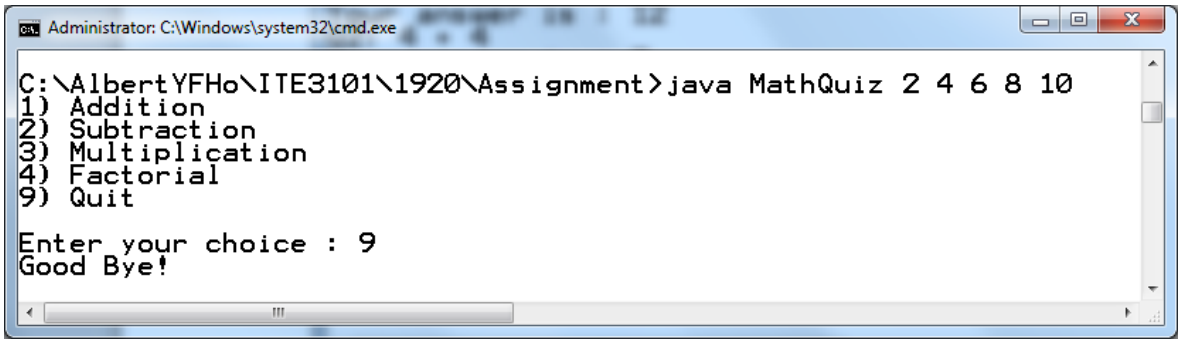
Note that each questions **should** have a ***question number***. For example, **Q1** for the first question and **Q5** for the fifth question.



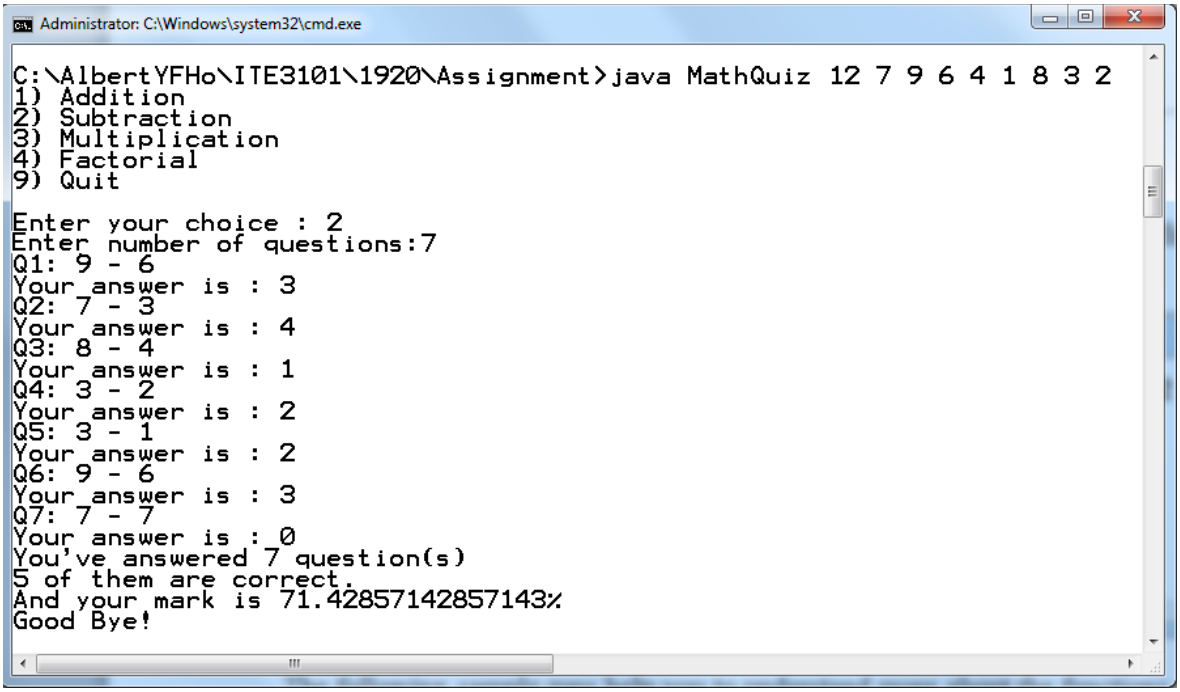
The result to be shown should contain (1) how many questions have been asked, (2) how many questions are answered correctly and (3) the percentage of correct answers. Lastly it shows a message "Good Bye!" and the program ends.

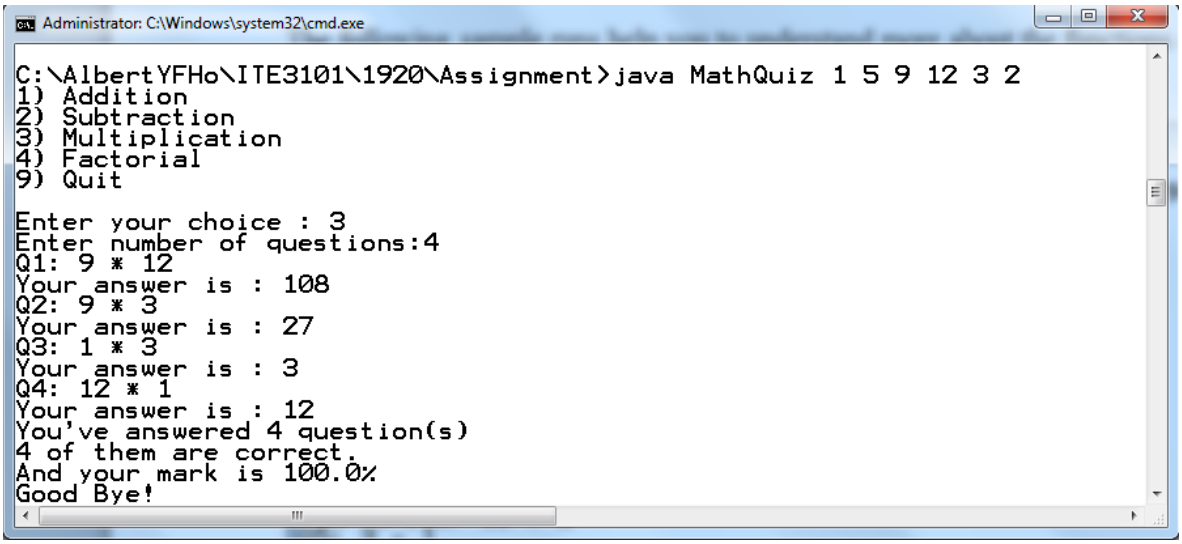
Step 4: Good Bye message

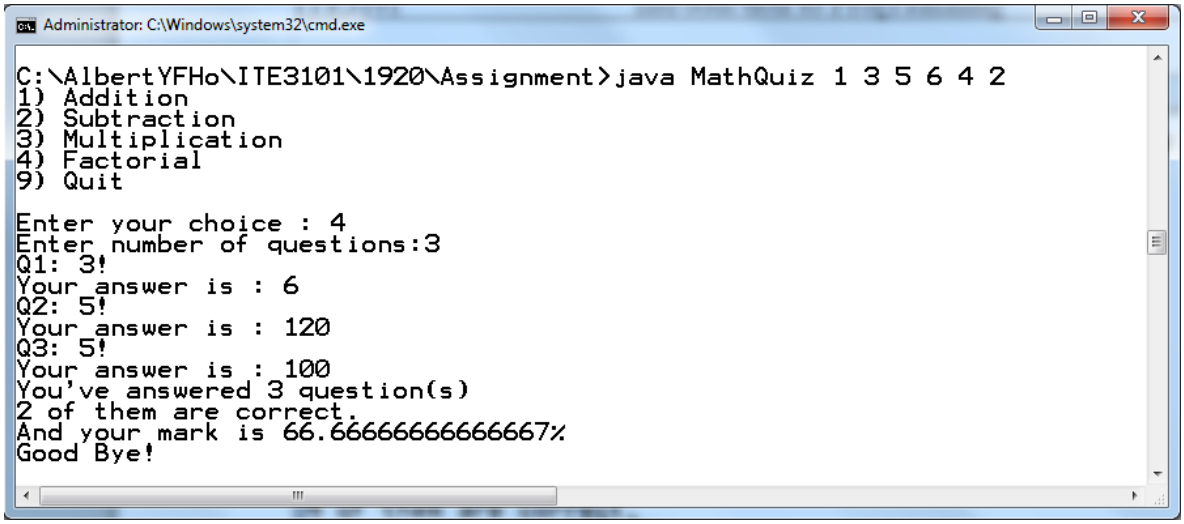
If *9* is entered in ***Step 1***, a message "Good Bye!" is shown and the program ends.



The following sample runs help you to understand more about the functions of the Mathematics Quiz System (apart from Addition shown above) :-







Assumption:

- You may assume the all user inputs are *valid* and no exception handling is required.
- All the input numbers in the command line of the *MathQuiz* program are positive integers.

Other Requirements:

You are required to design and implement a method to handle each of the following tasks:-

- Addition of 2 numbers
- Subtraction of 2 numbers
- Multiplication of 2 numbers
- Factorial of a number
- Get a number randomly from the list of numbers input in the command line (when the program runs)
- Calculate the number of questions answered correctly and show the result in command prompt

Assessment scheme

- Design and Implementation of the **SIX** methods specified in Other Requirements in the previous page. (30 marks)
- Algorithms for solving the problem (the whole program's skeleton). (20 marks)
- Correctness of the outputs. (30 marks)
- Programming style : naming, comments and indentation. (9 marks)
- Test Plan with test cases and test results. (11 marks)

~ END OF ASSIGNMENT ~