

Lab 4

1. Introduction to PASS (again)

PASS is the short name of Programming Assignment assessment System. In this course, you will use **PASS** for program testing and submission.

You may access **PASS** via the link in Canvas, or directly via <https://pass3.cs.cityu.edu.hk>
(You'll be using your CityU EID)


Select our course – **CS2311_JQ(2223 Sem.B) - Computer Programming**

To access problems, click the link "**Problem list**". 

Problems are divided into two types, namely *practice* and *assessment*. Assessment problems will be marked, and you **MUST** submit your solutions in order to be counted for grading. Practice problems are for exercise. You can test your solutions of practice problems on the PASS system, but no submission is required.

To test/submit a solution, click the "**Test/Submit**" icon  for the question you want to solve.

You may specify source code (.cpp file) upload with the "**Browse**" button (*default*), or you may paste the source code into the space provided. (*Need to select from radio button*)

To test the program (*practice or assignment*), click the **Test** button. 

*Note: Your program should follow the input and output format **EXACTLY** (i.e. identical spacing, new lines and letter case). Otherwise the **PASS** system will say that your program's output is incorrect.*

To submit for **assessment** problems, click the **Submit** button. Again, please be reminded that only the submissions via the **Submit** button are counted for grading. The code in **Test** will NOT be considered for grading. You are allowed to make multiple submissions to the same problem in the PASS system. However, please be noted that the TAs will grade and judge late submission based on your **very last** submission.

*Note: After submission, **PASS** will report the output of your program versus the "expected output". Note that for assignments, the test cases for "**Test**" may not be the same as what we use for grading. (Test cases for "**Test**", which you can see when you click the **Test** button, are usually a subset of the complete test cases we use for grading.)*

Sample problem. A sample problem is available on PASS to help you get familiar with the PASS system. You can find it in the problem list as **Lab03_3b** and test your solution on PASS. This question is the exercise we discussed in the previous lab.

2. Problems

NOTE: In all the following problems, the input entered by the user is highlighted by underline. It is not part of the output from the program.

Q1. [For Practice]

Write a program that calculates the result of 'a', 'operator', 'b' which are entered by users, like '1+4 = 5'.

- a) Verify whether the input 'a' and 'b' are digits.
- b) The operators include +, -, *, /, <, >, =.
- c) 'True' is simplified as 'T' while 'False' is simplified as 'F'.
- d) When the operator is '=', output '==' instead of '=' and add brackets to the equation, e.g., (1==2)=F.

Expected Output:

Example 1	Example 2
Enter the equation: <u>1</u> <u>+</u> <u>4</u> 1+4=5	Enter the equation: <u>10</u> <u>/</u> <u>6</u> 10/6=1.66667
Example 3	Example 4
Enter the equation: <u>a</u> <u>+</u> <u>1</u> Invalid input.	Enter the equation: <u>1</u> <u><</u> <u>4</u> 1<4=T
Example 5	Example 6
Enter the equation: <u>1</u> <u>\$</u> <u>4</u> Invalid operation.	Enter the equation: <u>5</u> <u>=</u> <u>5</u> (5==5)=T

Hint: Try to use switch .. case.

Q2. [For Assessment. Will Be Marked.]

You must click submit to submit your solution. Deadline is 11:59pm, Feb 14, 2023.

Write a program that calculates your age in future.

The console's prompts for data entry and the output displayed should strictly follow the format of the expected output shown below.

- a) The program should cope with singulars and plurals properly in the output, e.g. "1 month" but "2 months", and "0 month" or "0 year" can be in the output.
- b) The program should check for invalid month input. If an input number for month is not in the range from 1 to 12, the program should print the message below and end.

Invalid Month Input!

- c) The program should check for invalid age input.
 - 1. If an input number for age is a negative integer, the program should print the message below and end.
The value for age cannot be a negative integer!
 - 2. If an input number for present age is larger than 200, the program should print the message below and end.
Sorry, people may be dead by this age!
- d) The program should check that you were not born yet in the year and month for which you wish to know your age. The program should print the message below and end.

You were not born!

- e) Given the calculated age X year(s) and Y month(s), the program should check the followings:
 - 1. If both X and Y are odd, print the following message.
Both of X and Y are odd!
 - 2. If either X or Y is odd, print the number that is odd. For example, if X is odd, print the following message.
X is odd!
 - 3. If neither X nor Y is odd, print the following message.
Both of X and Y are not odd!

Hints: *Verify the inputs before calculating the outputs.*

Expected Output:

Example
Enter the current year: <u>2019</u> Enter the current month: <u>10</u> Enter your current age in years: <u>20</u> Enter the month in which you were born: <u>6</u> Enter the year for which you wish to know your age: <u>2030</u> Enter the month in the year for which you wish to know your age: <u>8</u> Your age in 2030/8: 31 years and 2 months 31 is odd!

NOTE: Your program MUST follow the EXACT input/output format! Otherwise, you may not pass the test cases even though your calculation is correct.