

# CSCI1300: Problem Set 1

Due: June 13, 2021 at 11:59 pm (MT)

Interview Grading Required

## Pseudocode

Consider the Homework Point Sum problem below. The problem states:

Get all homework scores for one student, calculate, and display the sum of the scores. Additionally, you should also display the word "Fail" if their average score is lower than 33. NOTE: In some cases, we may not know how many homework scores each student will enter. Also, you may safely assume that the homework scores are within the range [0,100].

Here are three algorithms that could solve the problem above:

A. Ask the user how many homework scores they would like to input. Then ask for a homework score - one at a time - for as many times as indicated by the user. Use the entered information to display the sum, check if the user failed and print "Fail" if they have.

Let us look at a possible sample run for algorithm A. The user-provided input is in **bold blue**:

```
Welcome to the Homework Point Sum program!
How many assignments did you complete?: 5
Enter score 1: 30
Enter score 2: 25
Enter score 3: 14.5
Enter score 4: 42
Enter score 5: 23
Homework point sum is 134.5
Fail
```

B. Ask the user for homework scores, one at a time, and have the user enter "-1" when there are no more scores to input, then use the entered information to display the sum, check if the user failed and print "Fail" if they have.

Let us look at a possible sample run for algorithm B. The user-provided input is in **bold blue**:

```
Welcome to the Homework Point Sum program!
Enter a homework score, or -1 to quit:
33
33
-1
Homework point sum is 66.0
```

C. After each score is entered, ask the user if they have more scores to enter. If they say "yes", ask them to enter the next score. If the answer is neither "yes" nor "no", print the following message "Unrecognized entry! Please try again." If they say "no", then use the entered information to display the sum, check if the user failed and print "Fail" if they have.

Let us look at a possible sample run for algorithm C. The user-provided input is in **bold blue**:

```
Welcome to the Homework Point Sum program!
Do you want to enter a homework score? ('yes' or 'no')
yes
Enter score: 20
Do you want to enter a homework score? ('yes' or 'no')
yes
Enter score: 25
Do you want to enter a homework score? ('yes' or 'no')
what
Unrecognized entry! Please try again.
Do you want to enter a homework score? ('yes' or 'no')
yes
Enter score: 30
Do you want to enter a homework score? ('yes' or 'no')
no
Homework point sum is 75
Fail
```

### Task 1 – 10 points

Write different sample runs for algorithms A, B and C that follow their respective format as the examples above.

(Because of code runner, please only vary the user input in bold blue from the examples above)

Submission format: Plaintext file (.txt) named `samplerun.txt`. Write both the sample-runs in the same file.

### Task 2 – 30 points

Write a pseudocode for algorithms A, B, and C. Make sure that your pseudocode logic matches the sample runs above.

Submission format: Plaintext file (.txt) named `pseudocode.txt`. Write all 3 pseudocodes in the same file.

## C++ Implementation

Now it is time to implement the above pseudocodes in C++. [Note: You may not be able to completely attempt portions of this task until after Lecture 5.](#)

### Task 3 – 60 points

Write three separate programs that implement algorithms A, B, and C in C++.

Submission format: Name these files `algA.cpp`, `algB.cpp`, and `algC.cpp`. Your program outputs must exactly match the above sample runs for each algorithm.

### Task 4 (Extra Credit) – 15 points

Write a program to get the average score (single number, may NOT be fractional) from the user. Calculate and display the letter grade to the user based on the following grading scheme (point ranges are inclusive):

Points Range	Grade
93-100	A
90-92	A-
87-89	B+

83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
67-69	D+
63-66	D
60-62	D-
0-59	F

Let us look at a possible sample run for the extra credit question. The user-provided input is in **bold blue**:

*Please enter your average score: **93***  
*Your grade is A*

Submission format: Name this file `extracredit.cpp`.

## Submission and Grading

### Submission format

Kindly submit a ZIP file with the required files on Canvas on or before the due date. Instructions for creating a ZIP file are available on Canvas (under Guides).

Name your folder and archive in the following format:

<LastName>\_<FirstName>.zip, i.e., if a student's name is Harry Potter, they should name their ZIP file: Potter\_Harry.zip.

Please submit the zip file containing your solutions here:

<https://canvas.colorado.edu/courses/73400/assignments/1050773>

For Tasks 3 (and Task 4 if you attempt it), in addition to including the solutions in your zip file, make sure to use the CodeRunner to run and submit your code.

The link for the CodeRunner:

<https://canvas.colorado.edu/courses/73400/assignments/1063270>

## Program Compilation notes

C++ programs will be compiled and executed using C++11. You may use the following command to compile and test your program (assumes terminal is in the same directory as the program):

```
g++ program_name.cpp -std=c++11 -o program_name  
  
./program_name
```

## Points Breakdown

The points breakdown is as follows:

Task	Points
Task 1: samplerun.txt	10
Task 2: pseudocode.txt	30
Task 3: algA.cpp, algB.cpp, algC.cpp	60
Total	100
Task 4: extracredit.cpp	15

## Interview grading

This assignment will be interview graded. Each task has a 50% weightage for your interview performance, i.e., if a task is worth 30 points, then 15 points is for your submission and 15 points for your interview performance.

Interviewer(s) will make their appointment slots available on Canvas' course calendar at least a day before the submission due date. It is the student's responsibility to schedule the appointment through Canvas' course calendar (available at top right on the Canvas course home page). If none of the appointment slots suit you, please email the course's email ID: [csci1300@colorado.edu](mailto:csci1300@colorado.edu) well ahead of time.

## Late Submission and Honor Code

Course's late submission policy will apply. University's honor code and course specific collaboration policy applies. Check the syllabus for details.