

Processing and displaying user inputs

Object Oriented Programming - Programming Coursework #1 (5 pts)

Due Date : April 30, 2020 5pm

1 Task

The objective of this coursework is to gain a hand-on experience on taking inputs from the keyboard and displaying numerical and character values. The students have to implement two programs. The first program (Program 1) takes from the keyboard, two numbers and prints out the average value of these two inputs (2.5 pts). The second program (Program 2) takes an alphabetic character and prints out the corresponding ASCII code as an integer (2.5 pts). The submitted codes have to satisfy the compilation requirements (Sec. 3.1): If the codes do not meet these requirements, the students will receive 0 point. We will test each program with five different user (keyboard) input combinations. If the submitted program generates correct answers for all five input instances, the students will receive 2.5 points per program (i.e., 5 points for two programs).

2 Inputs and outputs

For both Program 1 and Program 2, the input will be provided by the users using the keyboard. After taking the respective inputs, Program 1 will print out (i.e., display in the monitor) a real number, while Program 2 will print an integer.

2.1 Sample inputs and outputs

Program 1 input: The user will type two numbers separated by a space and then type 'Return' ('Enter').

```
1 2.0
```

Program 1 output to be visualized in the monitor:

```
1.5
```

Program 2 input: The user will type an alphabetic character and then type 'Return' ('Enter').

```
G
```

Program 2 output to be visualized in the monitor:

```
71
```

For this CW, your programs do not have to handle exceptions caused by incorrect input data types. For instance, for Program 1 the user wouldn't provide a character and a number:

```
c 2.0
```

3 What to hand in for assessment

Please submit the source codes for Program 1 and Program 2 and a brief report (a single report for Program 1 and Program 2) via Black Board. Comments in the codes and the report should be written in English. The report should provide code comments explaining the algorithms. If the submission does

not include a report, the maximum possible points will be kept at 40% of the full points. Please format the submission as a single zip file ‘StudentID-YourName.zip’ that includes the files of the source codes and the report,

e.g.,

20201111_KwangInKim.zip

- Program1.cpp

- Program2.cpp

- Report.pdf

The submitted source codes should contain ‘Program1.cpp’ and ‘Program2.cpp’ files each including the id and name of the submitting student.

In “Program1.cpp”

```
//StudentId YourName  
#include <iostream>  
Your code here ...
```

3.1 Compilation requirements and instruction

Each submitted code needs to be compilable by the GNU C++ compiler and in the Linux environment. Specifically, they should be **compilable and executable in UNI06 server**. Students can verify the correct compilation and execution of their codes by following the instructions provided in the accompanying ‘howtocompile.pdf’ document.