Homework 02

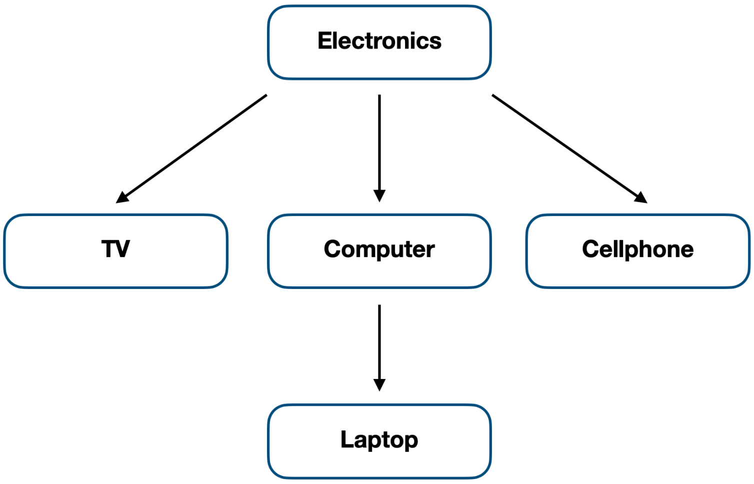
[2023-1] 데이터사이언스를 위한 컴퓨팅1(M3239.005500 001)

***Due: 2023*년 *4*월 *7*일 금요일 *23*시 *59*분**

1. C++ Programming Practice [50pts, NO partial credit]

Consider three different types of Electronics, TV, computer and cellphone. TV is the heaviest, and then computer and then cellphone.

Instruction:



1. The price is **size * x**, where **x=1.8, 0.6, 0.7** for TV, computer and cellphone, respectively.

2. Consider a laptop, which can have two GPU as max. When there are GPU, the price becomes twice.

3. Total price of laptop can be calculated as the sum of the price of laptop and GPU.

4. The Electronics class have

   a. `Size (member variable)`

   b. `GetSize`

   c. `SetSize`

5. Additionally, have

   a. `void PrintSelf(void)`

   b. `void AddGPU(Electronics *aGPU)` (laptop only, add GPUs)

6. `PrintSelf` prints

```
It is a {Electronic type}, Size={size}, Price={price}
```

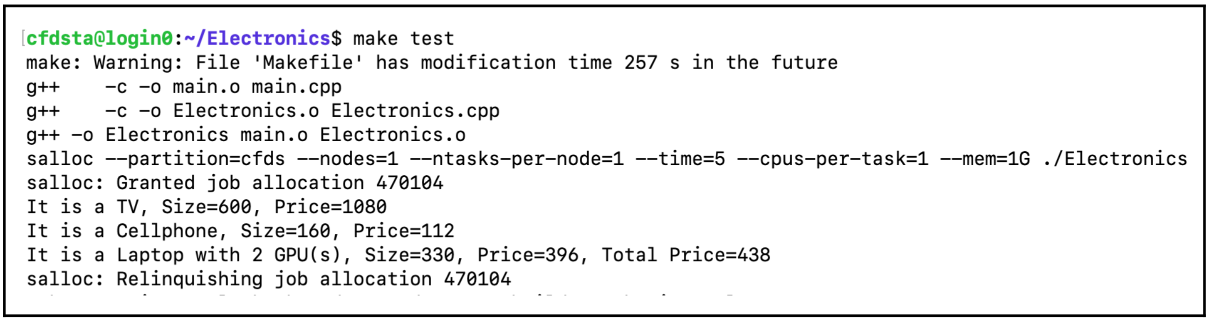
or (in the case of laptop)

```
It is a {Electronic type} with {# of GPUs} GPU(s), Size={size},
Price={price}, Total Price={totalprice}
```

**** Please omit all digits after the decimal point and print it out as shown in the example below, paying attention to spacing and capital letters.***

Implementation:

`Makefile` is provided to help compilation and job scheduling. Please compile and run it at GSDS server. You can run test code by running “**make test**”. Below is the results.



```
cfdsta@login0:~/Electronics$ make test
make: Warning: File 'Makefile' has modification time 257 s in the future
g++    -c -o main.o main.cpp
g++    -c -o Electronics.o Electronics.cpp
g++ -o Electronics main.o Electronics.o
salloc --partition=cfds --nodes=1 --ntasks-per-node=1 --time=5 --cpus-per-task=1 --mem=1G ./Electronics
salloc: Granted job allocation 470104
It is a TV, Size=600, Price=1080
It is a Cellphone, Size=160, Price=112
It is a Laptop with 2 GPU(s), Size=330, Price=396, Total Price=438
salloc: Relinquishing job allocation 470104
```

2. Word Counter [50pts]

Implement a WordCounter class which counts the number of word and characters from a text. The class should provide interfaces to count word `(GetWordCount)`, character `(GetCharacterCount)` and the number of unique words `(GetUniqueWordCount)`. In addition, it can count a particular word `(GetWordCount OneWord)`.

Instruction:

1. Treat upper and lower characters as different characters

    (so “He” and “he” are different word)

2. Do not include comma (,) and period (.) as characters.
3. Use std::string to store text.
4. The class can have the following member variable:
    a. `std::string text`
5. The class can have the following member functions:
    a. `InputText(std::string)`

    b. `int GetWordCount()`

    c. `int GetCharacterCount()`

    d. `int GetUniqueWordCount()`

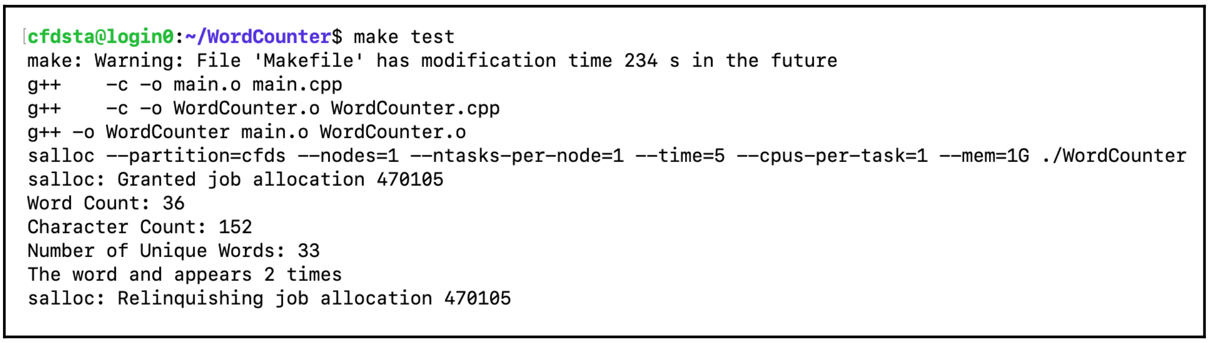
    e. `int GetWordCount_OneWord(const char *)`

    - Count how many times the word appears in the text.

Implementation:

Add all code related to class definition to `WordCounter.h` and all implementation to `WordCounter.cpp`. `main.cpp` have a test code.

`Makefile` is provided to help compilation and job scheduling. Please compile and run it at GSDS server. You can run test code by running “**make test**”. Below is the results.



```
[cfdsta@login0:~/WordCounter$ make test
make: Warning: File 'Makefile' has modification time 234 s in the future
g++    -c -o main.o main.cpp
g++    -c -o WordCounter.o WordCounter.cpp
g++ -o WordCounter main.o WordCounter.o
salloc --partition=cfds --nodes=1 --ntasks-per-node=1 --time=5 --cpus-per-task=1 --mem=1G ./WordCounter
salloc: Granted job allocation 470105
Word Count: 36
Character Count: 152
Number of Unique Words: 33
The word and appears 2 times
salloc: Relinquishing job allocation 470105
```

Additional explanation for input test case:

- Assume that only sentences with the correct spelling come in.
- A comma or period is always followed by a space.
- There is no case where a space comes before and after a comma or period at the same time. (There is no sentence like “There is only , in the sentence.”)
- There are no spaces at the beginning or end of a sentence.
- There is no case where two or more spaces occur in a row.
- No special characters other than commas or periods are not considered.
- The number of words in the input string does not exceed 1,000 words.
- Please refer to the example sentence in main.cpp and verify that the result is the same as the example run given in the problem.

3. Submission Instruction

- Compress Electronics directory and WordCounter directory as a single file and report it to ETL.

- In Electronics directory, **you cannot change main.cpp and Makefile**. You don’t need to submit them.

- In WordCounter directory, **you cannot change main.cpp and Makefile**. You don’t need to submit them.

- The file name you submit should be **YourAccount_HW02.zip**. (ex: cfds123_HW02.zip)

- **Any disadvantages caused by not referring to the submission instructions will be scored without exception.**

- Make sure your code works well on the GSDS server. Your code will be scored automatically by the program on the GSDS server. If you don’t follow the submission instruction, a penalty may occur.

- If you want to use your grace day, you must notify the TA by e-mail when submitting the homework. If you don’t notify, we will judge that you want to save your grace day for the next homeworks, so your homework is considered unsubmitted. Even if you use your grace day, your homework should be submitted through ETL.