Lab Test 5

Check Your Lab Test Environment

- Check whether your computer is working well.
- Print “Hello World!” in CLion.
- Please check the path “main.cpp” for the submission.

Announcement

- Cheating (including googling) is forbidden during the test. There will be a strong penalty if you're caught cheating.
- You can modify any part of the files if it meets the instructions.

Skeleton Code

- Download skeleton code for the problem 1.

Submission

- Zip “main.cpp” for problem 1 as “**20XX-XXXXX.zip**”.
- Upload the zip file to ETL.

Skeleton Code

The test ends at **19:15**

Submission

- Zip “main.cpp” for problem 1 as “**20XX-XXXXX.zip**”.
- Upload the zip file to ETL.

Problem 1 (C++) - Pills (1/3)

- In “main.cpp”, implement three subclasses of *Pill* (*SneezePill, RunnyNosePill, HeadachePill*) and *Patient::takeMedicine* method.
- Do not change the implementation of class *Pill* and main function in a skeleton code.

- Class *SneezePill*, *RunnyNosePill*, and *HeadachePill* should inherit from class *Pill*, and implement a public *take()* method.
  - *SneezePill::take()* should print a global string variable *SNEEZE_PILL*, and so on. Do **not** print newline (i.e. do not write “<< *endl*” in *take()* method)

Problem 1 (C++) - Pills (2/3)

- *Patient::takeMedicine(void *pointer, string type)*
  - *type* is one of “pill”, “pill address”, “3 pills”
- When type is “pill”
  - a void pointer *pointer* is actually *Pill* * type that points to exactly one Pill object. (To be more accurate, it is an object of one of three Pill subclasses)
- When type is “pill address”
  - A void pointer *pointer* is actually *Pill* ** type that points to a Pill pointer that points to exactly one Pill object.
- When type is “3 pills”
  - A void pointer *pointer* is actually *Pill* ** type that points to an array of 3 Pill pointers

Problem 1 (C++) - Pills (2/3)

- *Patient::takeMedicine(void *pointer, string type)* should call *take()* method of all Pill objects specified by *pointer* and *type*
  - That is, this method should call *take()* method of 1 Pill object when *type* is “pill” or “pill address”, and 3 Pill objects when *type* is “3 pills” **in exact order of the array**

- Not all test cases will check if the method works correctly for all 3 types, which will lead to partial score.

Expected Output

```
Patient patient;
SneezePill *sneezePill = new SneezePill;
RunnyNosePill *runnyNosePill = new RunnyNosePill;
HeadachePill *headachePill = new HeadachePill;

cout << "====Test Pill========" << endl;
patient.takeMedicine(sneezePill, type: "pill");
cout << "====================" << endl << endl;

cout << "==Test Pill Address==" << endl;
patient.takeMedicine(&runnyNosePill, type: "pill address");
cout << "====================" << endl << endl;

cout << "====Test 3 Pills=====" << endl;
Pill *pills[3] = { sneezePill, runnyNosePill, headachePill };
patient.takeMedicine(pills, type: "3 pills");
cout << "====================" << endl << endl;
```

```
====Test Pill========

No more sneeze

====================


==Test Pill Address==

No more runny nose

====================


====Test 3 Pills=====

No more sneeze

No more runny nose

No more headache

====================
```