



Setup HW05

ntut-xuan authored 1 month ago

Name	Last commit	Last update
Before_Make_A_Start.pdf	Setup HW05	1 month ago
README.md	Setup HW05	1 month ago

README.md

Homework 05

This homework was created by 黃漢軒 (109590031), also thanks to 賴奕廷 (Lai-YT) for reviewing the problem.

Please feel free to ask me if you have any questions.

Email: t10950031@ntut.org.tw / MS Teams 黃漢軒

⚠ Due: 11:59 p.m., day / month / 2022 ⚠

Goal

This homework has these goals:

- Know the concept of C++ virtual function.
 - Know the concept of C++ inheritance.
- Familiar with some built-in exceptions in C++ standard library.

I prepare a question set ([Before_Make_A_Start.pdf](#)) about these goals, which can help you to review these concepts.

Strongly recommend studying it before you make a start.

Folder Structure Tree

- You should finish the unit test written by you.
- You can split the unit test into multiple files, just remember to include all of it into `ut_main.cpp` (see course repo).

While your project has been built by `makefile`, the structure tree should be the same as the following section.

```
bin/  
├─ ut_all  
src/  
├─ alcohol.h  
├─ sour.h  
├─ tequila.h  
test/  
├─ ut_alcohol.h  
├─ ut_sour.h  
├─ ut_tequila.h  
├─ ut_main.cpp  
makefile
```

Problem Content

Uriah is operating a bar.

Uriah describes that alcoholic beverage should have their name, price, and alcohol content.

Moreover, Uriah provides a dilute function to let the customer dilute the alcoholic beverage by mixing some stuff.

Uriah will dilute or fortify the alcoholic beverage according to its type.

- If the alcoholic beverage is Sour, after the dilution, the alcoholic content will reduce by 30%.

- If the alcoholic beverage is Tequila, after the dilution, the alcoholic content will reduce by 5%.

In this task, you should complete three Classes: Alcohol in `alcohol.h`, Sour in `sour.h` and Tequila in `tequila.h`.

Furthermore, write the test case to test your code and make sure the code is working correctly.

See [the Task section](#) and [the Test section](#) below.

Task

You should complete the following class and its function, respectively.

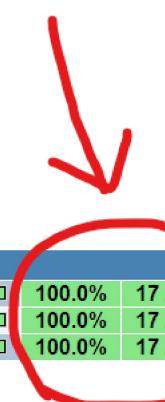
- Constructor
 - Alcohol
 - `Alcohol(string name, int price, double content)`
 - The parameterized copy constructor of the Alcohol class.
 - The name should not be empty (String length should not be zero), otherwise, throw an `invalid_argument` exception.
 - The price should not be negative, otherwise, throw an `invalid_argument` exception.
 - The content should be between `[0, 1]`, otherwise, throw an `invalid_argument` exception.
 - Sour
 - `Sour(string name, int price, double content)`
 - The parameterized copy constructor of the Sour class.
 - The name should not be empty (String length should not be zero), otherwise, throw an `invalid_argument` exception.
 - The price should not be negative, otherwise, throw an `invalid_argument` exception.
 - The content should be between `[0, 1]`, otherwise, throw an `invalid_argument` exception.
 - Tequila
 - `Tequila(string name, int price, double content)`
 - The parameterized copy constructor of the Tequila class.
 - The name should not be empty (String length should not be zero), otherwise, throw an `invalid_argument` exception.
 - The price should not be negative, otherwise, throw an `invalid_argument` exception.
 - The content should be between `[0, 1]`, otherwise, throw an `invalid_argument` exception.
 - Remember to add the default constructor and destructor, leave it blank if it isn't necessary.
- Function
 - Alcohol
 - `virtual void dilute() = 0`
 - Dilute the alcohol content, see [the problem statement](#).
 - `double get_alcohol_content()`
 - Return the content of the alcoholic beverage.
 - `int get_price()`
 - Return the price of the alcoholic beverage.
 - `string get_name()`
 - Return the name of the alcoholic beverage.
 - Sour
 - `void dilute()`
 - The virtual function of Alcohol.
 - Tequila
 - `void dilute()`
 - The virtual function of Alcohol.

- Operator overloading
 - Sour
 - Sour operator+(const Sour &other)
 - Combine two sours into one sour.
 - The name of the combined sour should be the name of the first sour.
 - The price of the combined sour should be the sum of two sours.
 - The content of the combined sour should be the average content of two sours.
 - bool operator==(const Sour &other)
 - The equals function of class Sour.
 - Sour &operator=(const Sour &other)
 - The copy assignment of class Sor.
 - Tequila
 - Tequila operator+(const Tequila &other)
 - Combine two tequila into one tequila.
 - The name of the combined tequila should be the name of the first tequila.
 - The price of the combined tequila should be the sum of two tequilas.
 - The content of combined tequila should be the average content of two tequila.
 - bool operator==(const Tequila &other)
 - The equals function of class Tequila.
 - Tequila &operator=(const Tequila &other)
 - The copy assignment of class Tequila.

Test

In this homework, you should use `gcovr` tool to make sure your *code coverage* in `/src` is all above 95%.

- If your lines of *code coverage* are below 95%, you will receive `FAILURE` in the HW Job.



File	Lines	Functions	Branches
alcohol.h	100.0% 17 / 17	100.0% 7 / 7	100.0% 15 / 15
sawa.h	100.0% 17 / 17	100.0% 6 / 6	100.0% 8 / 8
tequila.h	100.0% 17 / 17	100.0% 6 / 6	100.0% 8 / 8

****You will get the 35% score if HW Job passed, otherwise, you will lose the 35% score if HW Job failed. ****

See the course slide (`00P_gcovr.pptx`) to know how to install and how use it.

Notice

- Use [nullptr](#) if you want to have a null pointer, which is a special pointer that doesn't point to anything.
- Use `ASSERT_EQ` to test integers, `ASSERT_NEAR` to test floating-point numbers, and `ASSERT_THROW` to test exceptions.
- You should neither add a bin folder to your git nor add a file with the name `.gitignore` in the bin folder (see our class repo).
- In some situations you will lose score:
 - **You lose 5 points for each test that has a memory leaks. You can check memory leak with `valgrind` cmd.**

```
valgrind --track-origins=yes --leak-check=all <executable_file>
```
 - **You will lose 10% if your bin folder contains compiled ut_all in the git repo.**

