







### Change function return-type

ntut-xuan authored 2 months ago

Name	Last commit	Last update
 <code>src</code>	<a href="#">Change function return-type</a>	2 months ago
 <code>test</code>	<a href="#">Add some notice in goal section, upload ho...</a>	2 months ago
 <code>README.md</code>	<a href="#">Update README.md</a>	2 months ago
 <code>meme.jpg</code>	<a href="#">Setup HW#3</a>	2 months ago

### `README.md`

## Homework 03

This homework and content were created by 黃漢軒 (109590031), please feel free to ask me if you have any questions.

Email: [t10950031@ntut.org.tw](mailto:t10950031@ntut.org.tw) / MS Teams 黃漢軒

⚠ Due: 11:59 p.m., 21 / 10 / 2022 ⚠

### Goal

This homework has these goal:

- Know the conception of class, attribute and behavior.
- Know how to write basic constructor, destructor and member function (getter/setter).
- More familiar with how to use Git. **(If you still don't know how to use git, it will be very dangerous to take midterm.)**

The things you should complete:

- ☐ Write the `makefile` by yourself.
- ☐ Complete the function, constructor, destructor in `drink.h`
- ☐ Write some testcase on `ut_drink.h`
- ☐ Solve memory leak problem if it exists.

### Folder Structure Tree

**It's not necessary to remove all of the file in your repository, only need to update `ut_main.cpp` and `makefile`.**

I recommend to do that, because it can keep your repository clean :)

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

```
bin/
├─ ut_all
src/
├─ drink.h
test/
├─ ut_drink.h
├─ ut_main.cpp
makefile
```

### Problem Content

Uriah is a NTUT student from Tainan, you have been familiar with him in previous homework.

He love to eat cookie, but actually, he love eat all of the food contain sugar, like drink, candy, ..., etc.

Today, he want to build a *FoodPolarBear* to let the people order the food in online.

Because Uriah is busy in coding another part of project, you, as a best friend of Uriah, is asked to write a Class named *Drink* to help him record drink order.

In the task he gave you, you should finish 5 functions in *Drink* Class, see [the Functions section](#) in below.

## Functions

In this task, you should finish 5 function.

**You should use the source code we gave to finish this task.**

- `Drink::getName`
  - Return the name of drink.
  - If name is empty, you should **throw a string exception**.
- `Drink::getSweetnessLevel`
  - Return the level of sweetness.
  - If sweetnessLevel isn't set, you should **throw a string exception**.
- `Drink::addTopping`
  - Add a topping to drink.
- `Drink::getToppingByIndex`
  - Get the topping of drink by index, **the index is determine by the order of topping added and start from 0**.
  - If the index user query is out of range, you should **throw a string exception**.
- `Drink::getToppingCount`
  - Get the count of topping in drink.

For more information, see [the Sample section below](#).

Some important notice:

- The length of the name should be more than 4, the sweetness level should be between  $[0, 1]$ .
  - If user inputs an invalid value, you should throw a string exception.
- You can customize the attribute in *Drink* Class, just check the function can satisfy the requirement.
- You should also finish the constructor `Drink::Drink()` and destructor `~Drink::Drink()`.
- You should deal with the issue of memory leak when it happen.

## Sample

Assume we have a variable `drink`, which is constructed by the Class *Drink* you implement.

### Sample #1

```
/* Setup name to "Signature Black Tea with Milk" (熟成歐蕾). */
/* Setup sweetness Level to 0.3 */

drink.getName(); // Assert it will return "Signature Black Tea with Milk".
drink.getSweetnessLevel(); // Assert it will return 0.3.
```

### Sample #2

```
/* Add some topping */
drink.addTopping("Bubble");
drink.addTopping("Pudding");

/* Get the topping by index */
drink.getToppingByIndex(1); // Assert it will return Pudding.
drink.getToppingByIndex(0); // Assert it will return Bubble.
drink.getToppingByIndex(2); // Assert it will throw exception because we don't have the third topping.
```

### Sample #3

```
/* Add some topping */
drink.addTopping("Bubble");
drink.addTopping("Pudding");
```

```
/* Get the count of topping */  
drink.getToppingCount(); // Assert it will return 2.
```

## Test

You should write 12 test case to get the HW part of score (35%), and passed TA test to get TA part of score (65%):

**If you test the specific function, and write the test case that use another function, it cannot be said that you test two function in one test case.**

- HW Part of Score
  - [5%] Write two test case to check `Drink::Drink()` work correctly.
  - [6%] Write two test case to check `Drink::getName` work correctly.
  - [6%] Write two test case to check `Drink::getSweetnessLevel` work correctly.
  - [6%] Write two test case to check `Drink::addTopping` work correctly.
  - [6%] Write two test case to check `Drink::getToppingByIndex` work correctly.
  - [6%] Write two test case to check `Drink::getToppingCount` work correctly.
- TA part of Score
  - [5%] Your programming can pass test in sample section.
  - [60%] Your programming can pass test in the secret testcase which TA prepare.

## 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 integer, `ASSERT_NEAR` to test floating-point number, `ASSERT_THROW` to test exception.
- Please implement your test cases reasonably, otherwise you will get no point for the task.
- You should neither add bin folder to your git, nor add a file with the name of '.gitignore' in bin folder (see our class repo).
- Some situation you will lose score:
  - **You lose 5 points for each test that has memory leak. You can check memory leak with `valgrind` cmd.**
  - **You will lost 10% if your bin folder contains compiled ut\_all in git repo.**
- If you see segmentation fault, you can use `gdb` cmd to help debug. [Link](#)

**Ah the Friday is  
comming, so we can get  
some fun in the weekend...**

