Creative Software Programming, Assignment 12-1

Handed out: Dec 1, 2021

Due: 23:59, Dec 1, 2021 **(NO SCORE for late submissions!)**

- *Only files submitted by **git push to this course project at** [**https://hconnect.hanyang.ac.kr**](https://hconnect.hanyang.ac.kr) (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git) will be scored.*
- *Place your files under the directory structure **<Assignment name>/<Problem no.>/<your files>** just like the following example.*
  - ***DO NOT push CMake intermediate output files** (CMakeCache.txt, cmake_install.cmake, Makefile, CMakeFiles/*).*
  - ***DO NOT use cmake_minimum_required()** command in CMakeLists.txt.*

```
+ 2020_ITE0000_2019000001
  + 2-1/
    + 1/
      - 1.cpp
      - CMakeLists.txt
    + 2/
      - 2.cpp
      - CMakeLists.txt
    + …
```

- *The submission time is determined not when the commit is made **but when the git push is made**.*
- *Your files must be committed to the **master branch**. Otherwise, it will not be scored.*
- *Your program should output correct results even for inputs other than those used in the example.*
- *Basically, assignments are scored based on the output results. If it is not possible to check whether a requirement is implemented because the output is not correct, no score is given for the requirement, even if it is implemented internally. However, even if the output result is correct, no score is given for a requirement if the internal implementation does not satisfy the requirement.*

1. Write a program that works as follows:
    A. Define a myswap() function template that swaps the values of two arguments of any arbitrary type.
    B. Take two references to the myswap() function template: myswap(T & a, T & b).
    C. Take a pair of integers, real numbers, words (strings) from the user and call myswap() for each pair of input, and print out the swapped results.
        i. Do not define multiple versions of myswap() to handle multiple data types.
    D. Input:
        i. Two integers
        ii. Two real numbers
        iii. Two words (strings)
    E. Output: The swapped result
    F. Files to submit:
        i. A C++ source file
        ii. A CMakeLists.txt to generate the executable

```
$ ./swap
1 2
After calling myswap(): 2 1
1.1 2.2
After calling myswap(): 2.2 1.1
aaa bbb
After calling myswap(): bbb aaa
$
```

2. Write a program that works as follows:
    A. Complete the following my_container.h to define MyContainer class template that can contain an array of any arbitrary type.

B.

```
#pragma once

template <class T>
class MyContainer
{
public:
    MyContainer(int len)
    {
        // Implement here
    }
    ~MyContainer()
    {
        // Implement here
    }
    void clear()
    {
        // Implement here
    }
    int size()
    {
        // Implement here
    }

    template <class U>
    friend std::istream& operator>> (std::istream &in, MyContainer<U> &b);
    template <class U>
    friend std::ostream& operator<< (std::ostream &out, MyContainer<U>
&b);

protected:
    T * obj_arr = NULL;
    int n_elements;
};

template<class T>
std::istream& operator>> (std::istream &in, MyContainer<T> &b)
{
// Implement here
}

template<class T>
std::ostream& operator<< (std::ostream &out, MyContainer<T> &b)
{
          // Implement here
}
```

C. Do not add any other member functions or member variables, do not change the member access modifiers (public, private).

D. Take a number of real numbers, words (strings), integers from the user to create and fill MyContainer<T> instances, and print out their contents.

   i. Use stream operators to fill MyContainer<T> instances and print out them.

ii. Do not define multiple versions of class MyContainer to handle multiple data types.

E. Input:

i. Number of real numbers

ii. Real numbers

iii. Number of words

iv. Words

v. Number of integers

vi. Integers

F. Output: MyContainer instances

G. Files to submit:

i. main.cpp - main() must be in this file.

ii. my_container.h – Complete the given code skeleton

iii. A CMakeLists.txt to generate the executable

```
$ ./container
2
1.1 3.14
1.1 3.14
4
aaa bbb ccc ddd
aaa bbb ccc ddd
3
1 2 3
1 2 3
$
```