창의적 소프트웨어 프로그래밍 Lab 17

Handed out : Thu, Nov 24, 2022

Due : Mon, Nov 28, 2022, 23:59 (NO SCORE for late submissions!)

Submit your file on LMS.

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

```
$ ./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.

```
#pragma once

template <class T>
class MyContainer
{
public:
    MyContainer(int size)
    {
        // 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
}
```

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

C. Take a number of integers, real numbers, words (strings) 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.

D. Input:

   i. Number of integers

   ii. Integers

   iii. Number of real numbers

   iv. Real numbers

   v. Number of words

   vi. Words

E. Output: MyContainer instances

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