Creative Software Design, Assignment 5-1

Deadline: 2024-10-02 23:59 (No score for late submission)

- Submit your homework by uploading your zip file to the LMS assignment section. Below is an example.

```
13178_Assignment1-1_2024123456.zip
├─ 1.cc
├─ 2.cc
├─ 3.cc
└─ ...
```

- Your zip file name should follow this format: **13178_Assignment[Assignment-number]_[Student-ID].zip**
  - Ex. 13178_Assignment1-1_2024123456.zip

- Source files should be named as **<filename>.cc** *or* **<filename>.cpp**

- **You must submit your solution in the zip file before the deadline.**

1. Write a C++ program that retrieves a student’s score by their name.

   A. **Input and Functionality**

      1. The program first takes the number of students and then takes the name and score for each student.
      2. After storing the students’ names and scores, the program asks for a student’s name and retrieves their score.

   B. **Storage**

      1. You are required to store students’ names and scores using `std::map`.

   C. **Handling Non-Existing Names**

      1. If there is no student with the given name, the program should print a message similar to the one shown in Example 2.

   D. Example output of your program (Bold text indicates user input):

```
(example 1)
N =
5⏎
Tom 90⏎
Amy 80⏎
Bob 85⏎
Emma 95⏎
Aaron 95⏎
Which student's score?
Bob⏎
Bob's score: 85
```

```
(example 2)
N =
5⏎
Tom 90⏎
Amy 80⏎
Bob 85⏎
Emma 95⏎
Aaron 95⏎
Which student's score?
Nancy⏎
Nancy is not in the database.
```

   E. Submission file: one C++ source file (File name: 1.**cc** or 1.**cpp**)

2. Write a C++ program that merges two containers into one container.

   A. **Requirements:**

      1. **Function Definition:**

         - You must write functions to merge the following STL containers.

           i. std::vector<int>

           ii. std::list<int>

           iii. std::deque<int>

           iv. std::set<int>

           v. std::map<int, int>

      2. **Function Signature:**

         - Define a function:

           `void mergeContainers(T& container1, const T& container2)`

           where T can be one of the following types:

           i. std::vector<int>

           ii. std::list<int>

           iii. std::deque<int>

           iv. std::set<int>

           v. std::map<int, int>

      3. **Implementation Details:**

         - You can either:

           i. Write five different functions (one for each container) using function overloading.

           ii. Write one function using a template to handle all container types.

      4. **Function Behavior:**

         - After mergeContainers() is called, container1 should contain all the elements from both container1 and container2.

      5. **Handling Duplicate Keys (for `std::map`):**

         - If container1 and container2 have the same key, discard the element from container2.

   B. Submission file: one C++ source file (File name: 2.**cc** or 2.**cpp**)

      1. The file **must not** contain a `main()` function.

3. Write a C++ program that counts the appearance of words in a sentence.

   A. **Input**

      1. The program should take a sentence as input, such as in the example provided

   B. **Output**

      1. The program should output each word in the sentence along with the number of times it appears.
      2. The output order of the words does not matter.

   C. Example output of your program (Bold text indicates user input):

```
what are you doing doing how are doing do you wanna doing⏎
what: 1
are: 2
doing: 4
how: 1
you: 2
do: 1
wanna: 1
```

   D. Submission file: one C++ source file (File name: 3.**cc** or 3.**cpp**)