Creative Software Design, Assignment 13-1

Deadline: 2024-11-20 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. **Implementing Functions for Summation Using Modern C++ Features**

A. **Task Overview**

Write three C++ functions (or function templates) that take two parameters and return their sum using modern C++ features. You need to implement the following functions:

1. `add_int()`:

   i. Accepts **only two parameters of `int` type** and returns their sum.

2. `add_double()`:

   i. Accepts **only two parameters of `double` type** and returns their sum.

3. `add_any()`:

   i. Accepts **two parameters of any type** (including user-defined classes) and returns their sum.

   ii. For this function, ensure that the types of parameters allow addition (i.e., the operator+ must be defined for the types).

   - (Example) `add_any()` can take two `std::string` objects and return their concatenated result.

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

2. Implementing print_stats() Function Template for STL Containers

A. Task Overview

Implement a `print_stats()` function template that takes a sequence container from the Standard Template Library (STL) as its parameter. The function should calculate and print the following statistics of the numbers in the container:

Minimum, Maximum, Median, Average, Variance, Standard deviation

B. Requirements

1. **Accepted Containers**

   The parameter should be a sequence container from STL containing numeric types. Supported containers include:

   i. **`std::array`**

   ii. **`std::vector`**

   iii. **`std::deque`**

   iv. **`std::forward_list`**

   v. **`std::list`**

2. **Restrictions**

   i. You may only use **range-based `for` loops** for iteration.

   ii. Do not use traditional loops (e.g., `for`, `while`) or `goto` statements.

C. Submission file: one C++ source file (File name: 2.**cc** <u>or</u> 2.**cpp**)