CSC10003 – Object Oriented Programming

Week 09
Template

Cảm ơn thầy Trần Duy Quang đã cung cấp template cho môn học

Department of Software Engineering-FIT-VNU-HCMUS

1

Notes

Create a single solution/folder to store your source code in a week.

Then, create a project/sub-folder to store your source code of each assignment.

The source code in an assignment should have at least 3 files:

- A header file (.h): struct definition, function prototypes/definition.
- A source file (.cpp): function implementation.
- Another source file (.cpp): named YourID_Ex01.cpp, main function. Replace 01 by id of an assignment.
- Image of commit list

Make sure your source code was built correctly. Use many test cases to check your code before submitting to Moodle.

Name of your submission: **StudentID_W07_XX.zip**. XX: number of assignments you have done. XX: 00 – 99.

2

Content

In this lab, we will review the following topics:

- How to create your own class template

3

Assignments

A: YY = 01

H: YY = 02

3.1. Assignment MyVector

Define the following methods for your own class, MyVector, a class template.

In the main() function, create a vector of integers and a vector of fractions to test your defined methods.

Apply exception handling mechanism by throw exceptions in specific methods and try…catch… in the main() function.

```
template<class T>
class MyVector{
private:
        T *arr;
        int size;
public:
        // empty array
        MyVector();

        // n zeros
        MyVector(int n);

        MyVector(T *a, int n);
        MyVector(const MyVector &v);

        ~MyVector();

        int getSize();
        T getItem(int index);
        void setItem(T value, int index);

        void add(T value);
        void addRange(T *a, int n);
        void clear();
        bool contains(T value);
        void toArray(T *arr, int &n);
        bool equals(const MyVector &v);
        int indexOf(T value);
```

```
        int lastIndexOf(T value);
        void insert(T value, int index);
        void remove(T value);
        void removeAt(int index);
        void reverse();
        string toString();

        void sortAsc();
        void sortDesc();
};
```

3.2. Assignment 2 – 1 of SOLID Principles

The purpose of this assignment is to evaluate your understanding of the SOLID principles and how they can be applied in object-oriented programming. Students will demonstrate this understanding through a written report or a presentation.

For each of the five SOLID principles (Single Responsibility Principle, Open-Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, and Dependency Inversion Principle), students should include the following sections:

1. **Concept Explanation**
   - Define the principle in your own words.
   - Explain its importance in object-oriented design.
   - Use diagrams or analogies if necessary to illustrate the concept.
2. **Application in Assignments or Projects**
   - Reflect on the weekly assignments or projects you've completed in this course.
   - If you have applied the principle in your work, provide specific examples. Explain how your solution adhered to the principle and the benefits it provided.
   - If you have not applied the principle, identify potential opportunities to incorporate it into your previous assignments or projects. Propose modifications to your code or design to demonstrate its application. Provide code snippets or UML diagrams as evidence.
3. **Connection to the Seminar**
   - Do your assigned design patterns relate to the principle? Give examples.