Bern University
of Applied Sciences

394661-FS2019-0 - C++ Programming I

EXERCISE-08

TABLE OF CONTENTS

1 Introduction

This exercise of 394661-FS2019-0 will focus on the basic concepts of using templates. Templates are a very powerful tool of the C++ language. Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type. A template is a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using template concept.

You will learn the following topics when completing this exercise:

- Writing a simple Class Template
- Template specialisation
- Using Templates

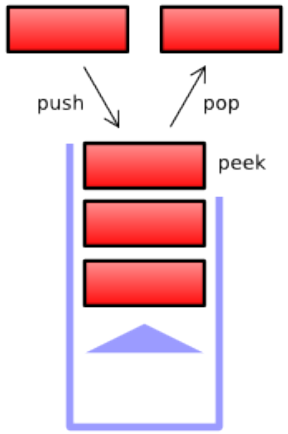
2 Exercises

Create CMake-Projects with C++ 11 compiler support and Debug/Release build options for the exercise. Add additional files manually to the project to gain full control over the included project files. Implement a **header only** version of the class template.

2.1 Implementation of simplified class `Stack`

Implement a simple class `Stack` holding a `std::array`, *e.g.* of size 10. A stack is a LIFO-buffer, `i.e.` last in - first out, which typically provides the following functionality:

- Add an element onto the top of the stack: `push()`
- Pop-off an element of the stack: `pop()` (return top element and delete from the stack, *i.e.* set to zero)
- Inspect the current element at the top of the stack: `peek()` (return top element)
- To keep track of the top use a private member `m_top`
- Specialize your class template to work with `std::string` to correctly handle the `pop()` function with an empty string
- Write a function `print` to display the 10 elements of the stack. Make sure `pop()` resets the elements correctly to it's default value.



Write your own test routine to test all three member functions of your implementation with `int`, `double` and `std::string`.

3 Code Review

Do a code review off one of your fellow student's code as demanded by ilias.

4 Submission

Submit your source code (as a zip-file) to Ilias EXERCISE-08 **before the deadline** specified in Ilias.