Bern University
of Applied Sciences

414078-HS2018-0 - C++ Programming II

EXERCISE-01

TABLE OF CONTENTS

1 Introduction

The first laboratory exercise of 414078-HS2018-0 will give a short insight of how to work with STL containers using iterators. In particular, the handy stream iterators is used to read data from `cin` or a file. In addition, you're invited to outline the first ideas for your code project. In order to write clear and readable code its crucial to read others peoples code! Therefore, find a code reviewing partner regardless of skill level differences.

The goal of this exercise is to:

- Use STL containers the "STL-way"
- Work with iterators and algorithms
- Outline the code project
- Find a code reviewer

2 Getting started with STL

Unless else specified, complete the following tasks uniquely using standard containers, algorithms and iterators. Don't use loops (for, while, do) or ordinary functions, indexing, etc. If you don't know the respective function or algorithm name search the web for help.

1. Write a program that creates a vector<char> which stores the letters a-z in order (Tip: `std::iota`). Print the elements to in the vector to `cout` in order and in reverse order using `std::copy` combined with the `ostream_Iterator`.
2. Create a `vector<string>` and read a list with names of car brands into the vector. Sort the list using std::sort and print the result using `std::copy`. There is a given text file with car brands, named *cars.txt*.
3. Rewrite exercise 2. Instead of printing all cars using `std::copy()`, construct a loop for printing all car brands in the list in exercise 2, which begins with a certain letter, e.g all car brands beginning with `'A'`. Try both

   a) the ordinary for loop, with iterators:
   ```
   for (auto it = cars.cbegin(); it != cars.cend(); ++it){...}
   ```

   b) and the range based for loop:
   ```
   for (const auto& car :  cars){...}
   ```

   Output is: `Adler Aero Alfa Anglo-Dane Audi Austin`
4. Rewrite exercise 2. Instead of printing all cars, print just those fulfilling a predicate. Define a binary predicate, which takes two strings and checks if the first character of the first string is equal to any of the characters in the second string, regardless of case. Use `std::copy_if` and your predicate to print all car brands beginning with, e.g. A, B or C. In this example, the output is:

   ```
   Adler Aero Alfa Anglo-Dane Audi Austin BMW Bentley Benz Borgwar Buick Cadillac
   Checker Chevrolet Chrysler Citroen
   ```

   Tipp: Use a lambda function to implement the binary predicate.

3 Filling Containers from `istream_iterators`

Write a program that reads user input from the command line and prints the unique words to the console. The program flow might look similar to:

```
Please enter some text:  a a b only unique words are are stored
The unique word list contains:

a are b only stored unique words
```

`Note:` Use `CTRL + D` to send a end-of-file character after your console input.

4 Code Project

The code project is a crucial part of this lecture. Therefore, we start early enough to develop project ideas:

- Think about a possible application you want to implement and outline your ideas in a few sentences. For example:
  - *My project is about capturing images from my laptop camera and do some fancy stuff with openCV, for example eye tracking and drawing glasses onto the faces (augmented reality)*
  - Take an algorithm from an other masters course (image processing / machine learning etc.), re-implement it in C++ and create a blazing fast application.
- Possible topics could be in the field of:
  1. Image or Signal Processing
  2. Augmented reality
  3. 3D volume rendering
  4. or any other field of your interest!
- The workload should be in the range of **20-30 hours**, that means it's a very small application with little features
- Choose a partner for the upcoming code reviews

5 Submission

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