Exercises:

1. Which statement is correct?
   a) In C++ compile time and run time constants can be defined with the keyword const.
   b) Because of type safeness it is not allowed to defined run time constants in C++.
   c) In C++, the keyword const is used for compile time constants; the keyword readonly is used to define run time constants.
2. When would you use pointers and when would you use references as parameters in a function? What are the advantages and disadvantages each?
3. Learn about the member functions of the STL type s*td::string.* Create a documentation that contains a summary of the member functions, of which you think they are important. Also show, how these member functions are typically used.
4. The user should input a string via command line arguments. The program checks, whether the passed string is a palindrome. A palindrome is a "symmetric" word, i.e. such a word that can be read from left to right and from right to left, e.g. "otto", "anna", "lagerregal". After the check following output should be generated: "anna is a palindrome". The case of the word's letters should be regarded, so "Otto" is no palindrome. The check for a word to be a palindrome should be implemented in a single function with following signature:

   ```
   bool IsPalindrome(const& std::string word);
   ```

   A function that tests an argument against a criterion is sometimes called predicate.
5. Develop a function that awaits a source string and a substring. The result of this function is the count of occurrences of the substring in the source string. This function will have following signature:

   ```
   int CountSubString(const& std::string source, const& std::string subString);
   ```

   The functionality of the function should be proved with a test program.

Remarks:

- Everything that was left unspecified can be solved as you prefer.
- In order to solve the exercises, only use known constructs, esp. the stuff you have learned in the lectures!
- The usage of goto, C++11 extensions, as well as #pragmas is not allowed. The usage of global variables is not allowed.
- **Please obey these rules for the time being:**
  - **The usage of goto, C++11 extensions, as well as #pragmas is not allowed.**
  - **The usage of global variables is not allowed.**
  - **You mustn't use the STL, because we did not yet understood how it works!**
  - **But *std::string*, *std::cout*, *std::cin* and belonging to manipulators can be used.**
- Only use classes for your UDTs. The usage of public fields is not allowed! The definition of inline member functions is not allowed!
- Do not put class definitions and member function definitions into separate files (we have not yet discussed separated compilation of UDTs).
- Your types should apply const-ness as far as possible. They should be const-correct. Minimize the usage of non-const&!
- The results of the programming exercises need to be runnable applications! All programs have to be implemented as console programs.
- The programs need to be robust, i.e. they should cope with erroneous input from the user.
- You should be able to describe your programs after implementation. Comments are

mandatory.

- In documentations as well as in comments, strings or user interfaces make correct use of language (spelling and grammar)!
- Don't panic: In programming multiple solutions are possible.
- Don't send binary files (e.g. the contents of debug/release folders) with your solutions! Do only send source and project files.
- If you have problems use the Visual Studio help (F1) or the Xcode help, books and the internet primarily.
- Of course you can also ask colleagues; but it is of course always better, if you find a solution yourself.