Institut für Automatisierungstechnik

C++ Basics and Applications in technical Systems

Exercise I (Lecture 1-3) in WiSe 2015/2016

Further information and this exercise as download on:
http://www.elearning.uni-bremen.de

Supervisors:
Adrian Leu - Santiago Focke - Björn Mindermann
{aleu, sfocke, bmindermann}@iat.uni-bremen.de

General comment: Always make sure to inform the user what he should do (use a menu with instructions) and to check if the user entered the correct data type, so that the program does not crash! (e.g. character, number).

1. Create a program that reads in two natural numbers and performs the following operations: (8)
   (a) Printing of the corresponding characters from the ASCII-table to the screen (see Note1).
   (b) Multiply the numbers with the fixed number 2 in two different ways (see Note2).
   (c) Perform the following division operations: Division, Modulo-Division and Division with real numbers (Typecasting from natural into real number. Don't use C-style typecasting! Use static_cast instead).

   Explain your results and print out all relevant steps.

   Notes:

   - **Note 1**: Don't forget to check if the user input is in a valid range, for which the characters can be printed out (e.g. 32...127).
   - **Note 2**: Use `*` and `<<` for multiplication!

2. Create a program for the calculation of a quadratic equation (e.g. using the quadratic formula). The program has the following functionalities: (13)
   (a) When a user presses the key `<r>` or `<R>`, the program will print out
   `"Please input the coefficients"`.
   The coefficients of the quadratic polynomial should be read.

(b) When a user presses the key `<c>` or `<C>`, the program will print out
"Quadratic equation will be calculated, the results are: "
and the results will be given. The real as well as the complex solution should be offered (see Note).

(c) When a user presses the button `<w>` or `<W>`, the program will print out
"The program is waiting".
Within the wait state the program has to execute a loop that terminates after a counter reaches a predetermined limit. This limit has to be set by the user. Please check the plausibility of the input (only non-negative limit is allowed) and print an error message if necessary.

(d) When a user presses the button `<t>` or `<T>`, the program will print out
"The program will be terminated".
Then the program terminates.

Notes:

- **Note 1**: Use the data-type complex
- **Note 2**: Be sure to treat all special cases of the equation

3. Create a program for the administration of a "book database". The database should be able to provide the following information for each registered book: (25)

   - Author's first name
   - Author's last name
   - Title of the book
   - Publisher
   - Price
   - Range of topics, for which only the following possibilities are available:
     - Chemistry
     - Physics
     - Electrical-engineering
     - Computer Science
     - Software
   - Reading state (specifies if the book has been already read by its owner [`true`/`false`]

   The user should be able to perform the following activities:

   (a) New book entry
   (b) Show book entry
   (c) Change book entry
   (d) Search a certain last name
   (e) Save the search results in a file
   (f) Program termination

   At the end of the program, the database should be cleared completely.

   Use the following data-types for the implementation of the program:

   - Enumeration type (enum)
   - Structured data-type (struct)
   - Data-types from the standard library (vector and string)

HowTo: Send in the Exercises

(Exercises are only accepted if they meet the following conditions)

- Consulting with colleagues is allowed, but copying is **not** permitted! Solutions will be accepted no later than the announced deadline.
- Special formatting requirements: Use spaces and tabs to align the lines of code within command blocks and use comments to explain what you did. Non-readable or not documented code will be severily graded down! If you use Qt, you can select all the code with Ctrl+A and then conveniently press Ctrl+I for automatic indentation. For more details about our guidelines see **IAT programming guidelines**.
- zip/rar the project-folder(s) and send everything in one file via Email or hand it in via usb-stick. Before compressing, delete the object-files (*.o) and the debug/release folders that contain the executable files. **Additionally a paper printout has to be delivered to the supervisors**.
- Preferably write your programs with Qt Creator. If you do not use Creator, you have to provide the source code (again delete object-files and executbale files) together with a working Makefile. If you use CASE-tools (like Rhapsody or Bouml) send in a full set of model-files **and** the generated C++ source code.