TCC CSCI 2843

Assignment 6

1) For this assignment, you will need to create two projects:
   a) A static library project.
   b) A console application that will link in the library from (1.a).
2) In the **library** project, place the files attached to this assignment:
   - trim.h
   - trim.cpp

   Compile this project to make sure that it produces a library (.lib file) and note the directory where it places the library. Also, note the directory where the source files reside (to know where the .h is). You will need these two directories to fill out the project settings in the application.
3) Create a separate **console application** project that will contain your code that will use the library. Make all the project settings necessary to point to the correct locations for **both** the **preprocessor** and the **linker** (for the header file(s) and library, respectively)
4) In the console application project, create an account.h (and account.cpp if necessary) to implement a class, account. An account needs to hold the following:

   std::string account code;
   std::string first_name;
   std::string last_name;
   double balance;

   Provide all necessary constructors, accessors, and operators.
5) Create a main.cpp in the console application project and include the account.h header (from your library project) in addition to <vector> and any other standard headers you will need (i.e <iostream> and <string>).
6) In the main function:
   a) Create a std::vector of accounts.
   b) Open a std::ifstream on the file "account.dat", which is in the following **line-oriented**, **fixed-column width** format:

      account_code: 10 characters
      first_name: 15 characters
      last_name: 25 characters
      balance: 8 digits, decimal place, 2 digits

c) In a loop, read each account, which is on a separate line (delimited with '\n'). The recommendation is to read each line as a std::string (e.g. with std::getline) and perform substrings to get the individual pieces of information. There are other possible implementations. If there are any errors either accessing the file or in the data format, **throw** an appropriate exception. Make sure you check for lines that are not the correct size (esp. too short)!!

Note - because each field is fixed-width, the first and last names can and will have blanks on the end. Use the "trim" or "trim_right" function (from the library) to remove the spaces from the names.

d) For each account read from the file, store it in the vector<account> created in (6.a).

e) Output the vector of accounts to a file, "account.csv", using a std::ofstream in comma**-delimited** format. For instance,

```
1234567890Fred          Murtz                 00002000.01
```

in the fixed-length file becomes

```
1234567890,Fred,Murtz,2000.01
```

7) For **all** exceptions, make sure an error message detailing the cause of the exception outputs to the console before exiting the program. Any further information (i.e. file and line number of exception) is purely optional.

8) For **ten bonus points**, use std::sort to sort the accounts by account number before writing them to the CSV file.