BEAR Bank | CSC232 Final Project | Due Date: December 5, 2020 at 11:59pm.

You will develop software for Bear Bank in C++ judiciously applying the data structures introduced in the CSC232 class while developing this Bear Bank software. The standalone software must run on Windows or MAC computers and offer the following features with flexibility for its users offering an intuitive console-based application.

1. [5 points] The banking system will have three types of users – system administrators, bank officials, and account holders.

2. [5 points] As of today, Bear bank offers three types of accounts – a) daily checking account with no monthly fees and no interest; b) daily savings account with no monthly fees and a variable daily interest rate set by the bank officials; c) fixed term, fixed-rate certificate of deposits (CD). Early cancellation of CD will incur a penalty based on the number of days remaining and no interest will be paid; d) Bear Bank management may decide to offer new account types with a monthly fee, fixed or variable interest rate, and penalty/service fees.

3. [10 points] Only the bank officials can – a) open or close an account; b) any closed account information must be saved with abilities for information retrieval (account number, account holder information, opening and close date, name and id of the bank official involved) by the bank officials; c) deposit or withdraw money from an account with the customer's permission, i.e. the customer must also enter the login password to authorize any transaction; d) search account(s) by account number, customer name or phone number.

4. [5 points] Each account must have the following information: a) account holder's first name and last name, phone number, address; b) interest rate, c) if applicable, then the term length, date of opening, closing, and maturity; d) total $ available; e) records of all transactions with dates, f) a unique account number.

5. [5 points] Only the system administrators can do the following – a) create and enable/disable bank official login profiles; b) create, delete, and modify account types in Requirements 2 and 3 above. c) retrieve any user's login id and change the password.

6. [10 points] A valid Bear bank customer – a) can have more than one account; b) change their login password; c) must log in with their user id and password to see the account(s) information, d) will see the last login date and time after every successful login, e) can check all transactions made with their account within a day/date range.

7. [10 points] Basic security and privacy must be considered for this software, such as a) username, password, account-related information, interest rate, and other settings, etc. must not be saved in plain text; b) administrators and bank officials must not be able to retrieve any user passwords.

Rubric: Total of 125 points.

1. Bear Bank software (core requirements 1-7): 50 points.
2. Documentation (see #4 in submission instructions): 30 points.
3. Code quality, handling of invalid inputs, how-to instructions: 30 points

4. Effective use of an online tool/platform for version control and collaboration: 5 points
5. Zoom project presentation (15 minutes, on Dec. 8, 9am-1pm, by appointment): 10 points

Note: The instructor may decide to award a maximum of 25 bonus points to outstanding submissions.

Submission Instructions:

1. Students will work on this project in a group consisting of 2-3 members.
2. Submit all codes and documents in a zip file in Blackboard.
3. Your submitted code files must be self-contained. It must compile and run on either Windows or Mac computer.
4. In a professionally formatted Microsoft Word file, you will provide -
    a. A description of the implemented features, including the core features and any additional features. Your interpretation of the provided features should be captured here.
    b. Detailed information on various data structures used to develop this application.
    c. A summary table consisting of all the data structures used, a brief description of the respective contexts, and a reference (e.g. code file names, class/function name, line numbers, etc.) to its implementations.
    d. In the body of the document (outside the table), for each of the data structures (including user-defined types and abstract data types) in each of the instances, you will justify the use of it with appropriate complexity analysis and any alternatives.
    e. The following information in appropriate locations:
        i. Project name, team members' information (name, email), semester.
        ii. Online repository information – provide a link to your online repository with the details visible to the instructor and the course teaching assistant.
        iii. Detailed instruction to setup, compile/run, and use your program.
5. It is highly expected that you will organize the contents of your project document in appropriate sections/sub-sections with a cover page and table of contents.
6. You may include a paragraph stating why your submission deserves a bonus point.

Submission deadline: December 5, 2020 at 11:59pm.

Project Objectives:

1. Show appropriate applications of simple, dynamic, and advanced data structures, abstract data types, user-defined types and complexity analysis.
2. Showcase knowledge of coding in C++, Object-Oriented Programming, Standard template libraries.
3. Writing a technical document supporting the design decisions for a complete software application.
4. Introduction to software planning and version control in a collaborative setting.
5. Effective teamwork.

CSC232 Project Teams: Fall 2020

#1
Michael Gegg
Sahejbir Bhatia
Jabez Nuetey

#2
Christian Leslie
Ethan Dawley
Ethan Tanner

#3
Braden Mills
Aaron Gerbrandt
Keegan Maynard

#4
Zoheb Ullah Sohel
Cole Kassing
Jaron Ritter

#5
Riley Grotenhuis
Daelon Kingore
Ashton Barnwell

#6
Kelly Christensen
Robert Safford
David Harper

#7
Vitor Freitas
Hung Nguyen
Huy Huynh

#8
Zijing Zhao
Nathan Obert
Dominic Zucchini