CSC 232 Project 1 – Keegan Maynard and Nathan Obert

**OOP Concepts**

|  |  |
| --- | --- |
| Abstraction | There are several header files included in the project. This was done so that the .cpp file only gathers input from the user and distributes the information to the OS header file where the commands get implemented. |
| Encapsulation | The different aspects of bank accounts were divided into separate classes for organizational and implementational needs. Further, the shared characteristics of each type of account was combined into the parent class BankAccount. Each derived class then implemented the class-specific methods and members it requires to perform the necessary operations. |
| Inheritance | The parent class, BankAccount, was used to create the common members and methods. Non-account specific behaviors were implemented directly in the header file where BankAccount is stored. The derived class-specific behaviors were made to be purely virtual and therefore were implemented directly inside the derived classes’ header files. |
| Polymorphism | The only methods that incorporated polymorphism were the pure virtual functions. These methods were common to both types of accounts, but were implemented differently by account requirements. |

**Contributions**

|  |  |  |
| --- | --- | --- |
| Nathan O. | 60% | Largely responsible for the initial UML class diagram and handling how users interacted with the account objects. As such, he wrote the .cpp file and the OS header file. Additionally, he tweaked the class implementations to improve cohesion with the OS header and user input requirements. |
| Keegan M. | 40% | Largely responsible for the creating the parent class as per the initial UML diagram and suggesting changes to be added to the design of the parent class, which means he further refined the class diagram. Additionally, he implemented the derived classes with some help from the other member. |

**How To**

|  |  |
| --- | --- |
| File Name | Description |
| Bear Bank.pdf | This pdf file is a UML diagram used to display the members and methods of the header files and C++ files used in the project. |
| BankAccount.h | This header file is the parent class for the other header files, and it contains the common class members and methods needed for performing operations and storing information. |
| SavingsAccount.h | This header file is a child class of BankAccount.h and contains the members and methods used to perform operations and store information for the savings accounts created by the user. |
| CheckingAccount.h | This header file is a child class of BankAccount.h, and contains the members and methods used to perform operations and store information for the checking accounts created by the user. |
| OS.h | This header file is a child class of BankAccount.h and contains the members and methods used to unite the account header files into a working banking system. |
| BearBank.cpp | This C++ file contains the main loop that controls the user interface, and makes calls to the header files in order to use their operations. |
| TotallyNotBankInfo.txt | This text file is a document used to store user account information, and is updated as the user continues to perform operations. This file will not be initially included in the zip file, the program will create the file once BearBank is compiled and executed. |

To run the program, unzip the files into the This PC\[local disk]\cygwin64\home\[profile name] directory. Then, open Cygwin and type the following line:

g++ BearBank.cpp -o BearBank

and press the enter key. This compiles the .cpp file to create an executable. To run the executable file, titled BearBank, type the following line:

./BearBank

and press the enter key. This will cause the program to run. As inputs are executed by the program, the TotallyNotBankInfo.txt file will be created and added to the directory.