

CS2720—Fall 2009

Assignment 2

Due: November 6, 2009 11:59 pm

- Your assignment will be marked on program correctness and readability (including comments).
- Doxygen documentation should be used.
- Write a Makefile which can be used to compile all programs in this assignment.
- Each class should be implemented in its own `.h` and `.cc` files.
- Use `const` member functions when appropriate.

In this problem, you will add error detection and recovery to your program from Assignment 1. You may use your own solution or one of the posted solutions to Assignment 1 as a starting point.

1. From the standard exception class `runtime_error`, derive the following exception classes:

- `invalid_file`: thrown by the main program if the given file name does not specify a file that can be opened.
- `input_format_error`: thrown by `Customer::read` and `Account::read` (or one of the subclasses) when the user input does not satisfy the correct format. You may want to throw these from the `readExtra` and `writeExtra` functions as well. For the purpose of this assignment, only check that the format is correct. You do not need to check that the data is correct (e.g. negative balance, negative interest rate, etc.)

Note: when this exception is thrown, be sure that the object being read into is not modified. The object should only be changed if the `read` operation is successful.

Hint: you may wish to read into temporary variables first before overwriting data member in `read()` so that the object is not partially read when an error occurs.

You may need to implement copy constructor and assignment operator for the `Account` class. It may also be convenient if the `Customer` class has a pure virtual function `Customer *clone()` which returns a pointer to a copy of itself.

- `insufficient_funds`: thrown by `Account::withdraw` (or one of the subclasses) when the requested withdrawal will result in a negative balance. When such a request is received, the exception should be thrown without actually performing the withdrawal. Be sure to account (pun intended) for any withdrawal fee.
- `invalid_menu_choice`: thrown by the main program if the menu choice entered is not valid.
- `account_not_found`: thrown by the main program if there is no account that matches the account number entered by the user.

You may declare and implement all exceptions in one `.h` file.

2. Add error detection code in the appropriate routines (see above) to detect the errors and throw the appropriate exceptions.
3. Add `try-catch` blocks in the appropriate routines to catch the exceptions and handle them appropriately. Some of the possible actions are (not mutually exclusive):
 - produce an error message;
 - ask the user to enter a request/input again;
 - ignore the request;
 - rethrow the exception to the caller;
 - exit the program.