PROGRAMMING PROJECT #4: Saving Account

Purpose:

- Develop a program using the C++ class and inheritance
- Solve problems using Object Oriented Programming
- Use arrays, vectors, strings, and files if needed
- Manipulate data by calling object's methods.
- Use exception to handle errors

Description:

Write a C++ program named **"MyBank"** that provides the following options:

1. Open an account
2. Deposit
3. Withdraw
4. Show balance (including principal and interest)
5. Show all transactions
6. Set interest rate
7. Exit

At the beginning, the program asks the user for the password before proceeding. The correct password is **"abc123"**. If the user enters the wrong password 3 times, the program prints an error message and exit.

Explanation of options:

1. Open an account: the program asks the user to enter name, initial balance year open and interest rate. This option only works the first time (where is no account yet). If an account already exists and the user chooses this option, it's considered an error. The program should print an error message.

2. Deposit: the program asks the user for amount and check to make sure the mount must be > 0. Otherwise, it throws an exception. You must define a user-defined exception. The program needs to keep track of the time and amount the user deposits money. If there is no account yet and the user enters this option, it's an error.

3. Withdraw: the withdraw amount must be < balance. Otherwise, an user-defined exception is thrown. The program needs to keep track of the time and amount the user withdraws money. If there is no account yet and the user enters this option, it's an error.

4. Show balance: call toString() method of the SavingAccount to display balance. If there is no account yet and the user enters this option, it's an error.

5. Show all transactions: The transactions must be shown with nice formatted column headings as follows:

Transaction type	**Amount**	**Date/Time**
Deposit	$250.50	Wed Apr 25 16:17:04 2018
Withdraw	$50	Wed Apr 26 09:28:18 2018

   The deposit and withdraw transactions must be kept track of by the corresponding methods.

   Hint: You need to declare an array of struct in your class to keep track transaction. Each transaction is a struct.

Requirements:

- All data members in your classes must be declared as “private” and provide “getter/setter” methods for them.
- You must have 1 superclass and 1 subclass. You can use BankAccount and SavingAccount as the starting point. You class must provide **showTransanctions()** method to display all transactions.
- You must also declare a user-defined exception class and make use of it in handling bad or invalid input. You decide what built-in superclass to inherit from.
- Well structure, easy to understand with the proper use of reusable functions, methods and classes.
- Handle user errors with proper error messages
- There must be at least 1 function. The methods in Account class do not count as functions.