~~Bank Account~~

~~Account Number (int)~~

~~Balance (double)~~

~~Static next account number~~

~~Default constructor~~

~~All properties constructor~~

~~Setters/getters~~

~~Deposit~~ **~~method~~**

~~Withdraw~~ **~~method~~**

~~Static validation (return Boolean)~~

~~Overloaded operators~~

~~Savings Account~~

~~Interest Rate (double)~~

~~Default constructor~~

~~All properties constructor~~

~~Setters/getters~~

~~Apply Interest~~ **~~method~~**

~~Overloaded operators~~

~~Checking Account~~

~~Transaction count (int)~~

~~Default Constructor~~

~~All properties constructor~~

~~Setters/getters~~

~~Increase Transaction Count~~ **~~Method~~**

~~Cheque~~ **~~Method~~**

~~Apply fees~~ **~~method~~**

*Allows setting balance to a negative value, but had to set variable to “protected” instead of “private”*

~~Customer Class~~

~~Name (string)~~

~~Accounts (BankAccounts)~~

~~Number of Accounts (int)~~

~~Default constructor~~

~~All properties constructor~~

~~Setters/getters~~

~~Overloaded operators~~

Main method

* + Create one instance of every class
    - ~~Bank Account~~
    - ~~Savings Account~~
    - ~~Chequings Account~~
    - ~~Customer~~
  + ~~One customer with both chequing and savings account~~
  + ~~One customer with no accounts~~
  + Test each method once
    - Bank Account Methods:
      * ~~Bank Account << operator~~
      * ~~Bank Account >> operator~~
      * ~~BankAccount();~~
      * ~~BankAccount(all properties);~~
      * ~~setAccountNum();~~
      * ~~setAccountBalance();~~
      * ~~getAccountBalance();~~
      * ~~getAccountNum();~~
      * isValidAccountNum(int);
      * isValidBalance(double);
      * ~~deposit(double);~~
      * ~~withdraw(double);~~
    - Savings Account Methods:
      * ~~Savings Account << operator~~
      * ~~Savings Account >> operator~~
      * ~~SavingsAccount();~~
      * ~~SavingsAccount(all properties)~~
      * ~~setInterestRate();~~
      * ~~getInterestRate();~~
      * isValidInterest(double)
      * ~~applyInterest();~~
    - Chequings Account Methods:
      * ~~Chequings Account << operator~~
      * ~~Chequings Account >> operator~~
      * ~~ChequingsAccount();~~
      * ~~ChequingsAccount(all properties);~~
      * ~~getTransactionCount();~~
      * ~~cheque(double);~~
      * ~~applyFees();~~
      * ~~withdraw(double);~~
    - Customer Class Methods:
      * ~~Customer << operator~~
      * ~~Customer >> operator~~
      * ~~Customer();~~
      * ~~Customer(all properties);~~
      * ~~setName(string);~~
      * ~~setAccounts(BankAccount[], int);~~
      * ~~setSavingsAccounts(SavingsAccount[], int);~~
      * ~~setChequingsAccount(ChequingsAccount[], int);~~
      * ~~getName();~~
      * ~~getAccounts();~~
      * ~~getSavingsAccounts();~~
      * ~~getChequingsAccounts();~~
      * ~~getNumAccts();~~
      * ~~getNumSavingsAccts();~~
      * ~~getNumChequingsAccts();~~