NTUST OOP Midterm Problem Design

Subject: Bank Account

Contributor: 邱韋霖, 鄭永泰, 范茗翔

Main testing concept: Basic I/O、Class

Basics	Functions
■ C++ BASICS	□ SEPARATE COMPILATION AND NAMESPACES
■ FLOW OF CONTROL	□ STREAMS AND FILE I/O
■ FUNCTION BASICS	□ RECURSION
□ PARAMETERS AND OVERLOADING	□ INHERITANCE
□ ARRAYS	□ POLYMORPHISM AND VIRTUAL FUNCTIONS
■ STRUCTURES AND CLASSES	□ TEMPLATES
■ CONSTRUCTORS AND OTHER TOOLS	□ LINKED DATA STRUCTURES
□ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES	□ EXCEPTION HANDLING
	□ STANDARD TEMPLATE LIBRARY
□ STRINGS	□ PATTERNS AND UML
□ POINTERS AND DYNAMIC ARRAYS	

Description:

Define a class named BankAccount to simulate bank deposits.

- The class BankAccount has two variables:
  static total(int): store the total money amount of all accounts in the bank.
  balance(int): store the money amount of this account.

- The class BankAccount has two constructors:
  - **BankAccount()**: construct a BankAccount where the balance is 0.
  - **BankAccount (int input):** construct a BankAccount where the balance is **input**.

- You should implement the following functions:
  - **void withdraw (int output)**: withdraw money in the bank with the value of **output**.
  - **void save(int input)**: save money in the bank with the value of **input**.
  - **int getBalance()**: return the current **balance** of this account.
  - **int getAllMoneyInBank ()**: return the value of the **total** money amount in the bank.

Input:

The main() function in your submission will be replaced when judging.
You can use the main() function in "Other Notes" to test your program.
No inputs for this exercise.

Output:

The result of executing your program with the given main function.

Sample Input / Output：

Sample Input	Sample Output
No inputs	100 100 0 50 50

■ **Eazy,Only basic programming syntax and structure are required.**
□ **Medium,Multiple programming grammars and structures are required.**
□ **Hard,Need to use multiple program structures or complex data types.**

Expected solving time:

10 minutes

Other notes:

```
int main() {
	BankAccount bankAccount1(200), bankAccount2, bankAccount3(-100);
	cout << BankAccount::getAllMoneyInBank() << endl;
	bankAccount1.withdraw(100);
	cout << bankAccount1.getBalance() << endl;
	cout << BankAccount::getAllMoneyInBank() << endl;
	bankAccount2.save(50);
	cout << bankAccount2.getBalance() << endl;
	cout << BankAccount::getAllMoneyInBank() << endl;
	return 0;
}
```