**CPP Problem Design**

|  |
| --- |
| **Subject: BankAccount** |
| **Contributor: 邱韋霖, 鄭永泰, 范茗翔** |
| **Main testing concept: Basic I/O、Class**   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  ■ FLOW OF CONTROL  ■ FUNCTION BASICS  □ PARAMETERS AND OVERLOADING  □ ARRAYS  ■ STRUCTURES AND CLASSES  ■ CONSTRUCTORS AND OTHER TOOLS  □ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES  □ STRINGS  □ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  □ STREAMS AND FILE I/O  □ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **Description:**  Write a definition of a class named BankAccount that will be used to store and amount of money call balance which are integers. You will need to declare and implement the following things:   1. Create a constructor “BankAccount(x)” that set the balance with initial value x. also have a default constructor “BankAccount()”that set balance with initial 0. 2. save(x): A member function to save money in the bank with an amount by the argument. 3. withdraw(x): A member function to withdraw money in the bank with an amount by the argument. 4. getBalance(): a const inspector functions to retrieve the current balance of the bank. 5. A static variable named: allMoneyInBank that track the total amount of BankAccounts have store. 6. A static function named: getAllMoneyInBank() that return the value of allMoneyInBank.   Noticed that the balance of BankAccount can be negative number so as allMoneyInBank  **Input:**  Replace main.cpp  **Output:**  See sample output.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | #include "BankAccount.h"  int main(void) {  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;  system("PAUSE");  return 0;  } | 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:**  **Replace main.cpp.**  **Only include BankAccount.h in main.cpp.** |