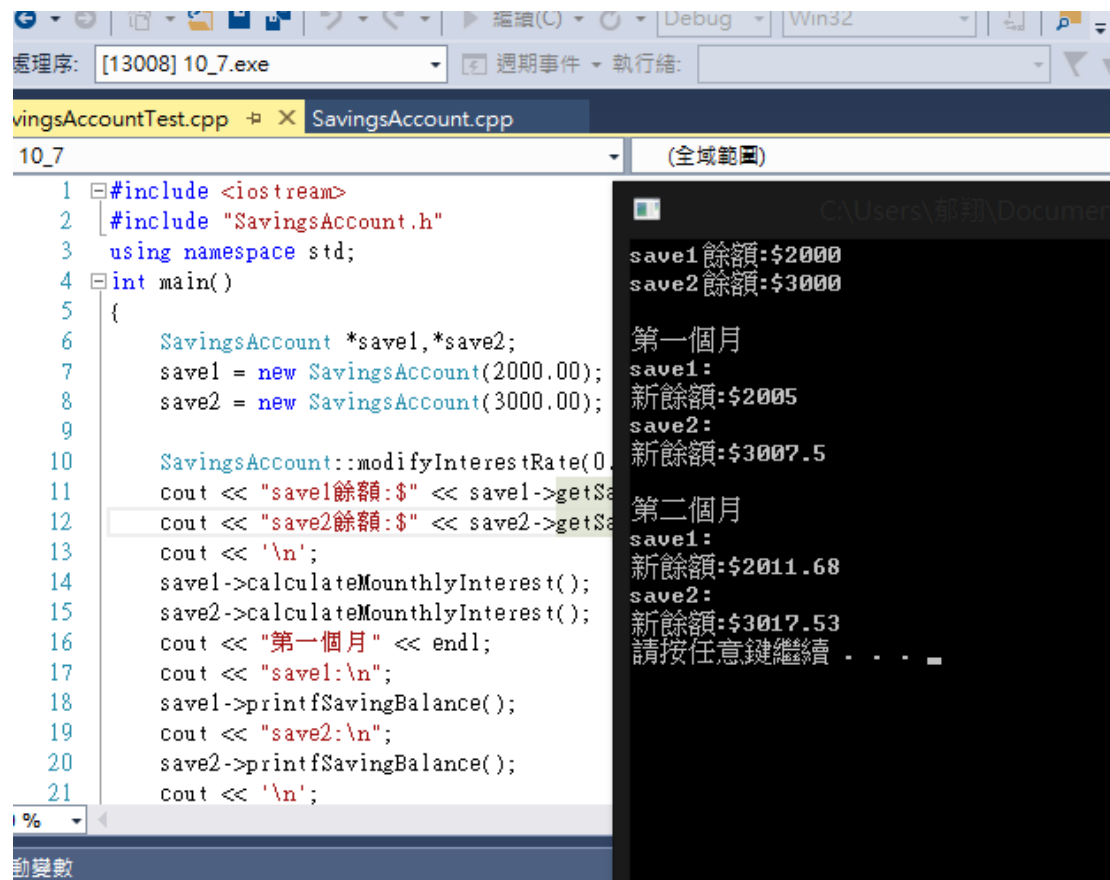


A3261576 林郁翔

10.7



The screenshot shows a C++ IDE with the file `SavingsAccountTest.cpp` open. The code defines a `SavingsAccount` class and a `main` function. The `main` function creates two `SavingsAccount` objects, `save1` and `save2`, with initial balances of 2000.00 and 3000.00 respectively. It then calls `modifyInterestRate(0.05)` on both accounts. The `main` function then prints the balances for both accounts, followed by the first month's interest calculation and the second month's interest calculation. The output window shows the results of the program execution.

```
1 #include <iostream>
2 #include "SavingsAccount.h"
3 using namespace std;
4 int main()
5 {
6     SavingsAccount *save1,*save2;
7     save1 = new SavingsAccount(2000.00);
8     save2 = new SavingsAccount(3000.00);
9
10    SavingsAccount::modifyInterestRate(0.05);
11    cout << "save1餘額:$" << save1->getSavingsBalance() << endl;
12    cout << "save2餘額:$" << save2->getSavingsBalance() << endl;
13    cout << '\n';
14    save1->calculateMounthlyInterest();
15    save2->calculateMounthlyInterest();
16    cout << "第一個月" << endl;
17    cout << "save1:\n";
18    save1->printfSavingsBalance();
19    cout << "save2:\n";
20    save2->printfSavingsBalance();
21    cout << '\n';
22}
```

Output:

```
save1 餘額:$2000
save2 餘額:$3000

第一個月
save1:
新餘額:$2005
save2:
新餘額:$3007.5

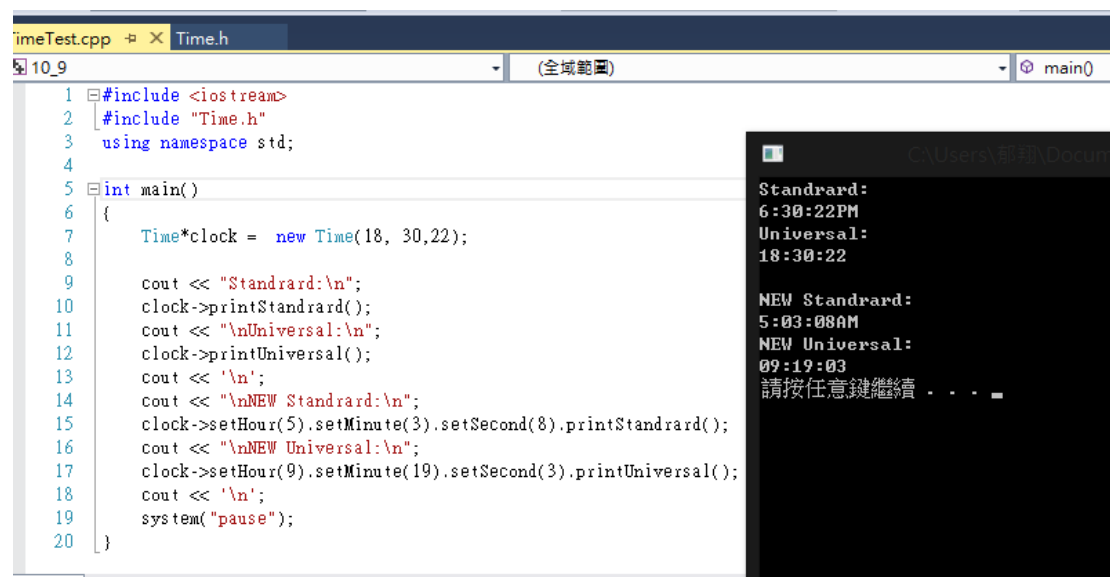
第二個月
save1:
新餘額:$2011.68
save2:
新餘額:$3017.53
請按任意鍵繼續 . . .
```

預設 Save1,Save2 餘額為 2000 及 300

再來經過第一個月的利息計算並印出新餘額

再來印下個月，即第二個月的利息計算並印出新餘額

10.9



The screenshot shows a C++ IDE with the file `TimeTest.cpp` open. The code defines a `Time` class and a `main` function. The `main` function creates a `Time` object, `clock`, with the time 18:30:22. It then prints the standard and universal times for `clock`. It then sets the time to 05:03:08 and prints the standard and universal times again. Finally, it sets the time to 09:19:03 and prints the standard and universal times. The output window shows the results of the program execution.

```
1 #include <iostream>
2 #include "Time.h"
3 using namespace std;
4
5 int main()
6 {
7     Time*clock = new Time(18, 30,22);
8
9     cout << "Standrard:\n";
10    clock->printStandrard();
11    cout << "\nUniversal:\n";
12    clock->printUniversal();
13    cout << '\n';
14    cout << "\nNEW Standrard:\n";
15    clock->setHour(5).setMinute(3).setSecond(8).printStandrard();
16    cout << "\nNEW Universal:\n";
17    clock->setHour(9).setMinute(19).setSecond(3).printUniversal();
18    cout << '\n';
19    system("pause");
20 }
```

Output:

```
Standrard:
6:30:22PM
Universal:
18:30:22

NEW Standrard:
5:03:08AM
NEW Universal:
09:19:03
請按任意鍵繼續 . . .
```

用 midNightSecond 當作午夜開始的秒數

一開始先設定 18 時 30 分 22 秒

測試 Strandard 時間

測試 Universal 時間

再來測試原 set 函式是否可執行，並串接 print 印出資料

測試把時間設為 5 時 3 分 8 秒印出 Strandard 時間

測試把時間設為 9 時 19 分 3 秒印出 Universal 時間