CPP 程式設計題

命題者：PWC

題目名稱(中文/英文)：namespace 練習(NameSpaceTraining)

主要測試觀念： namespace 應用

Basics		Functions	
☐	C++ BASICS 1	■	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
☐	STRINGS	☐	STANDARD TEMPLATE LIBRARY
☐	POINTERS AND DYNAMIC ARRAYS	☐	PATTERNS AND UML

題目說明： This Programming Project explores how the unnamed namespace works.

Listed below are snippets from a program to perform input validation for a username and password. The code to input and validate the username is in a separate file than the code to input and validate the password.

```
// File header user.cpp:
namespace Authenticate
{
   void inputUserName()
   {
      do
      {
        cout << "Enter your username (8 letters only)" << endl;
        cin >> username;
      } while (!isValid());
   }

   string getUserName()
   {
       return username;
   }
}
```

Define the **username** variable and the **isValid**() function in the unnamed namespace so the code will compile. The **isValid**() function should return true if username contains exactly eight **letters**. Generate an appropriate header file for this code.

Repeat the same steps for the file password.cpp, placing the password variable and the **isValid**()

function in the unnamed namespace. In this case, the **isValid**() function should return true if the input password has at least 8 **characters** including at least one non-letter:

```
// File header password.cpp:
namespace Authenticate
{
   void inputPassword()
   {
      do
      {
        cout << "Enter your password (at least 8 characters " <<
                    "and at least one non-letter)" << endl;
        cin >> password ;
      } while (!isValid());
   }

   string getPassword()
   {
       return password;
   }
}
```

At this point you should have two functions named **isValid**(), each in different unnamed namespaces. Place the following main function in an appropriate place. The program should compile and run.

```
int main()
{
   inputUserName();
   inputPassword();
   cout << "Your username is " << getUserName() << " and your password is: "
                                    << getPassword() << endl;
   return 0;
}
```

Test the program with several invalid usernames and passwords.

輸入說明：輸入帳號密碼

輸出說明：輸出帳號密碼

IO 範例 ：

	Sample Input	Sample Output
第一組測資與輸出	aaa aaaaaa	Enter your username (8 letters only) Enter your username (8 letters only)

	abcdefghi abcdefgh aaaaaaaaaa1	Enter your username (8 letters only) Enter your username (8 letters only) Enter your password (at least 8 characters and at least one non-letter) Your username is abcdefgh and your password is: aaaaaaaaaa1
…		

附屬資料：

☑解答程式：NameSpaceTraining.cpp(檔名), password.h, password.cpp, user.h, user.cpp

☑測試資料：input1.txt, input2.txt, input3.txt, output1.txt, output2.txt, output3.txt

■ 易，僅需用到基礎程式設計語法與結構

□中，需用到多項程式設計語法與結構

□難，需用到多項程式結構或較為複雜之資料型態或結構

解題時間：20 分鐘。

其他註記：

(1) 整體執行結果類似如下:

```
Enter your username (8 letters only)
aaa
Enter your username (8 letters only)
aaaaaa
Enter your username (8 letters only)
abcdefghi
Enter your username (8 letters only)
abcdefgh
Enter your password (at least 8 characters and at least one non-letter)
aaaaaaaaaa1
Your username is abcdefgh and your password is: aaaaaaaaaa1
```