

國立台灣海洋大學資訊工程系 C++ 程式設計實習 測試一

姓名：_____ 系級：_____ 學號：_____

96/03/20

考試時間：10:00 – 10:20

考試規則：1. 可以使用你的電腦

2. 不得左顧右盼、不得交談、不得交換任何資料、試卷題目有任何疑問請舉手發問

3. 繳卷時請繳交 簽名過之試題卷

多重選擇

1. 請問為什麼要將程式分散寫在多個檔案裡面?
 - a. 程式修改時不需要修改沒有關係的檔案
 - b. 程式偵錯時比較容易找到錯誤
 - c. 各個檔案可以獨立執行
 - d. 各個檔案可以獨立編譯
 - e. 可以用檔案來分割不同功能的模組
 - f. 可以在各個檔案裡使用相同名稱的變數和函式
 - g. 別人的程式都是這樣寫的
 - h. 其它_____

Sol: a, d, e (total 15 points, each addition takes 2 points off,
each missing takes 5 points off)

2. 下列關於 #include 的用法何者正確?
 - a. 每一個 .cpp 檔案裡面都應該加上 #include <iostream>
 - b. 如果 myClass.cpp 裡使用 cout 則 myClass.h 裡需要加上 #include <iostream>
 - c. 如果 myModule.cpp 裡有兩個函式 int a(int), double b(double) 則 myModule.h 裡一定要加上這兩個函式的宣告
 - d. myModule.h 檔案裡放的應該是 myModule.cpp 檔案裡讓其他模組使用的介面
 - e. 每一個 .cpp 檔案裡面應該需要加上它所使用的函式庫的 #include 敘述
 - f. 如果 .cpp 檔案裡面使用 string 類別應該要加上 #include <iostream>

Sol: d, e (total 15 points, each missing takes 5 points off,
each addition takes 2 points off)

簡答

3. 請完成下列 assert() 敘述?

```
bool inOrder(int numberOfElements, int array[])
{
    int i;
    for (i=0; i<numberOfElements-1; i++)
        if (array[i]>array[i+1])
```

```

        return false;
    return true;
}

// definition of function lessEqual

void main()
{
    int array[100];
    int numberOfData;
    ... // read in some integers from a file
    ...// store the number of data in the variable numberOfData
    qsort(array, numberOfData, sizeof(int), lessEqual); // sort the data in ascending order

    assert(inOrder(numberOfData, array)); // 10 points
    ...
}

```

4. 請完成下列 vector 的定義與使用

```

#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
struct Data {
    int iValue;
    double dValue[2];
};
void main() {
// 請以 vector 宣告 Data 型態的物件容器
vector<Data> dataContainer; // 5 points
    Data tmp; int i;
// 下列由 cin 讀入20組資料 10 1.5 1.3 12 1.4. 2.7 ... 順序儲存在 dataContainer 中
    for (i=0; i<20; i++)
    {
        cin >> tmp.iValue >> tmp.dValue[0] >> tmp.dValue[1];
        dataContainer.push_back(tmp);
    }
// 請寫一個迴圈將所有 dataContainer 內的資料反序寫入檔案 outfile 中
// 每一列存放一組資料
    ofstream outfile("data.txt");
    for (i=dataContainer.size()-1; i>=0; i++)
    {
        outfile << dataContainer[i].iValue << ' ' << dataContainer[i].dValue[0]
            << ' ' << dataContainer[i].dValue[1] << endl;
    }
}
}

```