

Turnin 作業 6

- Turnin Code: [cpp.hw6](#)
- Due Date: 4/27 Monday 23:59:00 (midnight) **Hard Deadline**

繳交方式說明

本次作業繳交將以資料夾的形式繳交，需要為每一題建立一個資料夾（資料夾名稱為該題目前方之代號，第一題為p1第二題為p2餘以此類推）。

繳交說明可參考[Turnin 作業 4](#) 中 p6 到 p10

任何未依照正確繳交格式的檔案將以 0 分計。



本文使用「」及「`\n`」代表「空白字元」與「Enter 換行字元」，並且將使用者輸入的部份使用灰階方式顯示。另外，题目的執行結果中，如果出現「(」、「)」、「:」、「;」、「.」與「,」等符號，皆為英文半形！

p1 計算表面積與體積

Jepsen 想要撰寫一個能夠計算圓柱體與正四角柱（底面為正方形的柱體）這兩種容器表面積以及體積的程式。她想要讓使用者先選擇想要計算的容器：若選擇圓柱體 `Cylinder` 程式會向使用者詢問底面圓形的半徑 `radius` 以及柱體高度 `height` 若選擇正四角柱 `Square prism` 則會向使用者詢問底面正方形的邊長 `side` 以及柱體高度 `height` 最後，請幫助 Jepsen 完成透過使用者輸入的內容計算出柱體的表面積與體積。

請參考下列的 `main.cpp` 程式（請依據檔名旁的路徑至 `ws` 取得程式碼）：

```
#include "container.h"
#include <iostream>
using namespace std;

double get_input_greater_than_0()
{
```

```
double number;
bool first_time = true;
do
{
    if (!first_time)
    {
        cout << "Value must be greater than 0!!! Please re-enter: ";
    }
    first_time = false;
    cin >> number;
} while (number <= 0);
return number;
}

int main()
{
    double input;
    Container container;
    Container *ptr = &container;
    do
    {
        cout << "Please select a container(1 for Cylinder, 2 for Square
Prism): ";
        cin >> input;
    } while ((input != 1) && (input != 2));

    container.type = (input == 1) ? CYLINDER : SQUARE_PRISM;

    if (container.type == CYLINDER)
    {
        cout << "Please input the radius of the base: ";
        container.radius = get_input_greater_than_0();
    }
    else
    {
        cout << "Please input the side length of the square base: ";
        container.side = get_input_greater_than_0();
    }
    cout << "Enter the height of the container: ";
    container.height = get_input_greater_than_0();

    calculate(ptr);

    cout << "surface area = " << container.surface_area << endl;
    cout << "volume = " << container.volume << endl;
    return 0;
}
```

請完成名為 **container.cpp** 與 **container.h** 的 C++ 語言程式，其中分別包含 calculate 函式的 **Implementation** 與其 **Prototype** 宣告

相關的使用者自訂資料型態請參考如下 required.h 檔案，請於 `container.h` 檔案中載入 required.h 標頭檔

```
#define PI 3.14

enum Type
{
    CYLINDER,
    SQUARE_PRISM
};

struct Container
{
    Type type;
    union
    {
        double side;
        double radius;
    };
    double height;
    double surface_area;
    double volume;
};
```

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp container.o
    g++ main.cpp container.o
container.o: container.cpp container.h
    g++ -c container.cpp
clean:
    rm -f *.o *~ *.*~ a.out
```

此題的執行結果可參考如下：

```
[3:23user@wsap1] ./a.out
Please select a container (1 for Cylinder, 2 for Square Prism): 1
Please input the radius of the base: 10
Enter the height of the container: 10
surface area = 1256
volume = 3140
[3:23user@wsap1] ./a.out
Please select a container (1 for Cylinder, 2 for Square Prism): 2
```

```
Please input the side length of the square base: 10
Enter the height of the container: 10
surface area = 600
volume = 1000
[3:23 user@wsapl]
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 請於 `container.h` 檔案中載入 `required.h` 標頭檔；圓周率請考慮 `required.h` 中定義的 `PI`
- 本題若有使用浮點數的需求，請使用 `double` 型態。
- 本題應繳交檔案如下（至於 `main.c` 與 `Makefile` 則不需繳交）：
 - `container.h`
 - `container.cpp`

p2 找出質數

請設計一個 C++ 語言程式，此程式將接收使用者輸入的一個整數 `n`，請配置一個大小為 `n` 的整數陣列，並找出前 `n` 個質數並存入該陣列中。

請參考下列的 `main.cpp` 程式（可依據檔名旁的路徑至 `ws` 取得程式碼）：

```
#include "prime_number.h"
#include <iostream>
using namespace std;

int get_input_greater_than_0()
{
    int number;
    bool first_time = true;
    do
    {
        if (!first_time)
        {
            cout << "Your input must greater than 0!!! Please re-enter: ";
        }
        first_time = false;
        cin >> number;
    } while (number <= 0);
    return number;
}

int main()
{
    int prime_num_quantity;
```

```
int *ptr;
cout << "How many prime numbers do you want to display? ";
prime_num_quantity = get_input_greater_than_0();

allocate_for_prime_num(&ptr, prime_num_quantity);

for (int i = 0; i < prime_num_quantity; i++)
{
    if (i != prime_num_quantity - 1)
    {
        cout << ptr[i] << ", ";
    }
    else
    {
        cout << ptr[i] << endl;
    }
}

delete ptr;
return 0;
}
```

請完成名為 **prime_number.cpp** 與 **prime_number.h** 的 C++ 語言程式，其中分別包含 `allocate_for_prime_num` 函式的 **Implementation** 與其 **Prototype 宣告**

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp prime_number.o
    g++ main.cpp prime_number.o
prime_number.o: prime_number.cpp prime_number.h
    g++ -c prime_number.cpp
clean:
    rm -f *.o *~ *.*~ a.out
```

此題的執行結果可參考如下：

```
[3:23user@wsap2] ./a.out ↵
How many prime numbers do you want to display? 10 ↵
2, 3, 5, 7, 11, 13, 17, 19, 23, 29 ↵
[3:23user@wsap2] ./a.out ↵
How many prime numbers do you want to display? 15 ↵
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47 ↵
[3:23user@wsap2] ↵
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題若有使用浮點數的需求，請使用 `double` 型態。
- 本題應繳交檔案如下（至於 `main.c` 與 `Makefile` 則不需繳交）：
 - `prime_number.h`
 - `prime_number.cpp`

*p3 通訊錄

請設計一個 C++ 語言的程式，能夠對通訊錄的資料進行增刪查改。

請參考下列的 `main.cpp` 程式（請依據檔名旁的路徑至 `ws` 取得程式碼）：

```
#include "contacts.h"

void help(){
    cout << "----- manual -----" << endl;
    cout << "a: add a contact" << endl;
    cout << "d: delete a contact" << endl;
    cout << "s: sort by name" << endl;
    cout << "f: find a contact" << endl;
    cout << "l: list all contact" << endl;
    cout << "q: terminate this program" << endl;
    cout << "h: show this manual" << endl;
}

int main()
{
    bool quit = false;
    char cmd = '\0';
    int data_count = 0;
    Contacts *data = nullptr;
    Contacts *ptr = data;

    while (!quit)
    {
        cout << ">> ";
        cin >> cmd;
        switch (cmd)
        {
            case 'h':
            {
                help();
                break;
            }
            case 'a':
```

```
        {
            add_a_contact(&data, &data_count);
            break;
        }
        case 'd':
        {
            delete_a_contact(&data, &data_count);
            break;
        }
        case 's':
        {
            sort_by_name(&data, &data_count);
            break;
        }
        case 'f':
        {
            find_a_contact(&data, &data_count);
            break;
        }
        case 'l':
        {
            list_data(&data, &data_count);
            break;
        }
        case 'q':
        {
            quit = true;
            cout << "Bye!" << endl;
            break;
        }
        default :
        {
            cout << "Unknown command. Please type \'h\' for help." <<
endl;
            break;
        }
    }
}
delete[] data;
return 0;
}
```

請完成名為 **contacts.cpp** 與 **contacts.h** 的 C++ 語言程式，其中分別包含 `add_a_contact`、`delete_a_contact`、`sort_by_name`、`find_a_contact` 以及 `list_data` 函式的 **Implementation** 與其 **Prototype** 宣告，函式的實作要求將詳述如下：

- `add_a_contact` 每新增一個聯絡人資訊，你需要新增一個 Contacts 的一個空間，向使用者取得「姓名」、「生日」、「手機號碼」以及「電子郵箱」，並將這些資訊存入該新增空間中（所有輸入內容均不需檢查其格式正確與否）。

- `delete_a_contact`
 1. 刪除一個聯絡人，你還需要釋放所配置儲存該聯絡人的記憶體空間。
 2. 若未找到要刪除的聯絡人，則輸出 `Contact not found!`
 3. 如果通訊錄已經沒有任何聯絡人了，則輸出 `Nothing to delete!`
- `sort_by_name`
 1. 將所有聯絡人姓名以字典序 `dictionary order` 的方式進行排序。
 2. 如果通訊錄已經沒有任何聯絡人了，則輸出 `Nothing to sort!`
- `find_a_contact`
 1. 根據使用者輸入的姓名找出通訊錄中姓名完全相同的聯絡人。
 2. 若未找到要刪除的聯絡人，則輸出 `Contact not found!`
 3. 如果通訊錄已經沒有任何聯絡人了，則輸出 `Nothing to find!`
- `list_data`
 1. 列表顯示通訊錄中所有內容。
 2. 如果通訊錄已經沒有任何聯絡人了，則輸出 `No contacts available.`

相關的使用者自訂資料型態請參考如下 `required.h` 檔案，請於 `contacts.h` 檔案中載入 `required.h` 標頭檔

```
#include <string>
using namespace std;

struct Contacts
{
    string name;
    struct Birthday
    {
        int year;
        int month;
        int day;
    } birthday;
    string phone_number;
    string email;
};
```

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp contacts.o
    g++ main.cpp contacts.o
contacts.o: contacts.cpp contacts.h
    g++ -c contacts.cpp
clean:
    rm -f *.o *~ *.~*~ a.out
```

此題的執行結果可參考如下：

```
[3:23user@wsap3] ./a.out
```

```

>>^l^
No^contacts^available.^
>>^a^
Enter^name:^Frank^
Enter^birthday(year^month^day):^3452^34^92^
Enter^phone^number:^0987654321^
Enter^email:^f rank@home^
Contact^added^successfully!^
>>^a^
Enter^name:^dennis^
Enter^birthday(year^month^day):^9234^23^12^
Enter^phone^number:^0912345678^
Enter^email:^dennis@wii.com^
Contact^added^successfully!^
>>^a^
Enter^name:^adam^
Enter^birthday(year^month^day):^8341^2^19^
Enter^phone^number:^8472940573^
Enter^email:^adam@adam^
Contact^added^successfully!^
>>^a^
Enter^name:^amy^
Enter^birthday(year^month^day):^8934^1^1^
Enter^phone^number:^9254431333^
Enter^email:^amy@school.tw^
Contact^added^successfully!^
>>^l^
^^Name^^^BirthDay^^^Phone^Number^^^Email^
[Frank][3452-34-92][0987654321][frank@home]^
[dennis][9234-23-12][0912345678][dennis@wii.com]^
[adam][8341-2-19][8472940573][adam@adam]^
[amy][8934-1-1][9254431333][amy@school.tw]^
>>^s^
Contacts^sorted^by^name!^
>>^l^
^^Name^^^BirthDay^^^Phone^Number^^^Email^
[Frank][3452-34-92][0987654321][frank@home]^
[adam][8341-2-19][8472940573][adam@adam]^
[amy][8934-1-1][9254431333][amy@school.tw]^
[dennis][9234-23-12][0912345678][dennis@wii.com]^
>>^d^
Enter^the^name^of^the^contact^to^delete:^emi ly^
Contact^not^found!^
>>^d^
Enter^the^name^of^the^contact^to^delete:^f rank^
Contact^not^found!^
>>^d^
Enter^the^name^of^the^contact^to^delete:^F Rank^
Contact^deleted^successfully!^

```



```

Enter the name of the contact to find: dennis
Name: dennis
Birthday: 9234-23-12
Phone Number: 0912345678
Email: dennis@wii.com
>>d
Enter the name of the contact to delete: dennis
Contact deleted successfully!
>>l
Name Birthday Phone Number Email
[Frank][3452-34-92][0912345678][frank@home]
>>d
Enter the name of the contact to delete: Frank
Contact deleted successfully!
>>f
Nothing to find!
>>d
Nothing to delete!
>>s
Nothing to sort!
>>q
Bye!
[3:23 user@ws p3]

```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 請於 `contacts.h` 檔案中載入 `required.h` 標頭檔
- 本題若有使用浮點數的需求，請使用 `double` 型態。
- 本題應繳交檔案如下（至於 `main.c` 與 `Makefile` 則不需繳交）：
 - `contacts.h`
 - `contacts.cpp`

p4 銀行帳戶類別實作

在我們日常生活中常常需要用手機 app 或是提款機查看帳戶裡的餘額、與朋友吃飯的時候會麻煩朋友先墊付之後再轉帳、現金不夠用的時候需要到 ATM 提款.....在本題我們將依照這些日常生活設計一個 `BankAccount` 類別 `Class`

請參考下列的 `main.cpp` 程式（請依據檔名旁的路徑至 ws 取得程式碼）：

```

#include "bank.h"

void show_div()

```

```
{
    cout << "-----" << endl;
}

int main()
{
    BankAccount Bob("Bob");
    BankAccount Justin("Justin");
    BankAccount Amy("Amy");

    show_div();

    Bob.show_balance();
    Justin.show_balance();
    Amy.show_balance();

    show_div(); // -----

    Bob.save(1945);
    Justin.save(476);
    Amy.save(1453);

    show_div(); // -----

    Bob.withdraw(1900);
    Bob.transfer(46, Justin);
    Bob.transfer(45, Amy);
    Bob.show_balance();

    show_div(); // -----

    Amy.show_balance();
    Amy.withdraw(15);
    Amy.show_balance();
    Amy.withdraw(2000);
    Amy.withdraw(1500);
    Amy.withdraw(1400);
    Amy.show_balance();

    show_div(); // -----

    Justin.transfer(1453, Amy);
    Justin.transfer(12, Amy);
    Justin.show_balance();
    Amy.show_balance();

    return 0;
}
```

請完成名為 **bank.cpp** 與 **bank.h** 的 C++ 語言程式，其中分別包含 BankAccount 類別以及其成員函式

show_balance, save, withdraw 與 transfer 的 **Implementation**

- 建構函式 BankAccount 參考主程式 10 ~ 12 行，建構函式的引數為姓名。
- 成員函式 show_balance 顯示帳戶當前餘額。
- 成員函式 save 對帳戶進行儲值。
- 成員函式 withdraw 對帳戶進行提款。
- 成員函式 transfer 轉帳給其他帳戶。

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp
    g++ main.cpp
clean:
    rm -f *.o *~ *.~ a.out
```

此題的執行結果可參考如下：

```
[3:23user@wsap4] ./a.out
-----
Bob's current balance: 0
Justin's current balance: 0
Amy's current balance: 0
-----
Bob saved $1945
Justin saved $476
Amy saved $1453
-----
Bob withdrew $1900
Bob doesn't have enough money to transfer!
Bob transfered $45 to Amy.
Bob's current balance: 0
-----
Amy's current balance: 1498
Amy withdrew $15
Amy's current balance: 1483
Amy doesn't have enough money to withdraw!
Amy doesn't have enough money to withdraw!
Amy withdrew $1400
Amy's current balance: 83
-----
Justin doesn't have enough money to transfer!
Justin transfered $12 to Amy.
Justin's current balance: 464
Amy's current balance: 95
[3:23user@wsap4]
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
 - bank.h

p5 學生類別實作

請撰寫一個 C++ 程式幫助助教計算全班的平均，並找出平均分數最高的學生。

請參考下列的 main.cpp 程式（請依據檔名旁的路徑至 ws 取得程式碼）：

```
#include "student.h"

int main()
{
    double class_average = 0;
    Student student[] = {
        {"Alex", "cbb999001", 82, 75, 93},
        {"Bob", "cbb999002", 40, 73, 96},
        {"Charlie", "cbb999003", 88, 76, 57},
        {"Dennis", "cbb999004", 82, 21, 93},
        {"Emily", "cbb999005", 82, 73, 76},
        {"Frank", "cbb999006", 90, 94, 98},
        {"George", "cbb999007", 82, 90, 93},
        {"Hawk", "cbb999008", 90, 70, 93},
        {"Iris", "cbb999009", 82, 73, 93},
        {"Jessica", "cbb999010", 82, 33, 97}
    };
    Student highest = student[0];

    for (int i = 0; i < 10; i++)
    {
        student[i].calculate_average();
        if (student[i].average > highest.average)
        {
            highest = student[i];
        }
        class_average += student[i].average;
    }

    class_average /= 10.0;
    cout << "Class Average = " << class_average << endl;
    cout << "The student with the highest average score: "
        << highest.name << "(" << highest.stu_id << "): " <<
highest.average << endl;
```

```
    return 0;
}
```

請完成名為 **student.h** 的 C++ 語言程式，其中分別包含 Student 類別中的成員函式 calculate_average 的 **Implementation**

- 成員變數 average 必須初始化為 0。
- 建構函式 Student 參考主程式 7 ~ 16 行，建構函式的引數從左到右依序為「姓名」、「學號」、「分數 1」、「分數 2」以及「分數 3」。
- 成員函式 calculate_average 計算該物件中三次考試的平均（無條件捨棄小數），並儲存在 this→average 變數中。

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp
    g++ main.cpp
clean:
    rm -f *.o *~ *.~* a.out
```

此題的執行結果可參考如下：

```
[3:23user@wsap5] ./a.out↵
ClassAverage=78.5↵
The student with the highest average score: Frank(cbb999006):94↵
[3:23user@wsap5]
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題若有使用浮點數的需求，請使用 double 型態。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
 - student.h

From:
<https://junwu.nptu.edu.tw/dokuwiki/> - Jun Wu的教學網頁
國立屏東大學資訊工程學系
CSIE, NPTU
Total: 275930

Permanent link:
<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=cpp:2026spring:hw6>

Last update: 2026/04/21 13:50



