Turnin 作業 7

- Turnin Code: cpp.hw7
- Due Date: 5/4 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「 」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，題目的執行結果中，如果出現「(」、「)」、「:」、「;」、「.」與「,」等符號，皆為英文半形！

---

p1 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 類別與其成員函式。由於物件導向特性的幫助，本題設計好的 BankAccount 類別將可以在不同的程式裡重複使用；因此本題將會提供兩個使用 BankAccount 類別的程式main1.cpp 與 main2.cpp並搭配 Makefile 將它們分別編譯成為 a.out-1 與 a.out-2 等兩個可執行檔案。要特別注意的是，本題批改時，還會使用其它的程式(例如main3.cpp、main4.cpp 等）進行編譯，以確認你所設計完成的 BankAccount 類別符合需求。

請參考下列的 main1.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;
}
```

請參考下列的 main2.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 Austin("Austin");
    BankAccount Jessica("Jessica");
    BankAccount Alison("Alison");

    show_div(); // ------------------

    Austin.show_balance();
    Jessica.show_balance();
    Alison.show_balance();

    show_div(); // ------------------

    Austin.save(1945);
    Austin.save(20000);
    Jessica.save(92456);
    Alison.save(453459);

    show_div(); // ------------------

    Austin.show_balance();
    Jessica.show_balance();
    Alison.show_balance();

    show_div(); // ------------------

    Austin.withdraw(1900);
    Austin.transfer(892314, Jessica);
    Jessica.transfer(84298, Alison);
    Alison.transfer(35545723, Austin);

    show_div(); // ------------------

    Austin.show_balance();
    Jessica.show_balance();
    Alison.show_balance();

    show_div(); // ------------------

    Alison.show_balance();
    Alison.withdraw(15);
```

```
    Alison.show_balance();
    Alison.withdraw(78245);
    Alison.withdraw(19855124);
    Alison.withdraw(19834565);
    Alison.show_balance();
    Alison.save(89412354);
    Alison.show_balance();
    return 0;
}
```

請完成名為 **bank.h** 與 **bank.cpp** 的 C++ 語言程式，它們分別是 BankAccount 類別的 **宣告定義** 與 **實作**

本題將使用以下的 Makefile 進行編譯：

```
all: main1.cpp main2.cpp bank.o bank.h
    c++ main1.cpp bank.o -o a.out-1
    c++ main2.cpp bank.o -o a.out-2
bank.o: bank.cpp bank.h
    c++ bank.cpp -c
clean:
    rm -f *.o *~ *.*~ a.out*
```

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p1]▲./a.out-1↵
----------------------↵
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↵
[3:23▲user@ws▲p1]▲./a.out-2↵
----------------------↵
Austin's▲current▲balance:▲0↵
Jessica's▲current▲balance:▲0↵
Alison's▲current▲balance:▲0↵
----------------------↵
Austin▲saved▲\$1945↵
Austin▲saved▲\$20000↵
Jessica▲saved▲\$92456↵
Alison▲saved▲\$453459↵
----------------------↵
Austin's▲current▲balance:▲21945↵
Jessica's▲current▲balance:▲92456↵
Alison's▲current▲balance:▲453459↵
----------------------↵
Austin▲withdrew▲\$1900↵
Austin▲doesn't▲have▲enough▲money▲to▲transfer!↵
Jessica▲transfered▲\$84298▲to▲Alison.↵
Alison▲doesn't▲have▲enough▲money▲to▲transfer!↵
----------------------↵
Austin's▲current▲balance:▲20045↵
Jessica's▲current▲balance:▲8158↵
Alison's▲current▲balance:▲537757↵
----------------------↵
Alison's▲current▲balance:▲537757↵
Alison▲withdrew▲\$15↵
Alison's▲current▲balance:▲537742↵
Alison▲withdrew▲\$78245↵
Alison▲doesn't▲have▲enough▲money▲to▲withdraw!↵
Alison▲doesn't▲have▲enough▲money▲to▲withdraw!↵
Alison's▲current▲balance:▲459497↵
Alison▲saved▲\$89412354↵
Alison's▲current▲balance:▲89871851↵
[3:23▲user@ws▲p1]▲
```



- 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.cpp 與 Makefile 則不需繳交）：
  - bank.h
  - bank.cpp

p2 學生類別實作

本題的目的是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** 與 **student.cpp** 的 C++ 程式，它們分別是 Student 類別的 **宣告定義** 與 **實作**。

- 成員變數 average必須將其數值初始化為 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）。

本題的相關程式將使用以下的 Makefile 進行編譯：

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

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p2]▲./a.out↵
Class▲Average▲=▲78.5↵
The▲student▲with▲the▲highest▲average▲score:▲Frank(cbb999006):▲94↵
[3:23▲user@ws▲p2]▲
```



- 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.cpp 與 Makefile 則不需繳交）：
    - student.h
    - student.cpp

p3 旅客類別實作

請撰寫一個程式幫助鳳梨航空寄出關於航班起飛前相關資訊的 email 給旅客。

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 "tourist.h"
using namespace std;

int main()
{
    Tourist Dennis("Dennis", "Smith");
    Dennis.flight.number = "CI 950";
```

```
    Dennis.flight.departure = "Kaohsiung (KHH)";
    Dennis.flight.destination = "Hong Kong (HKG)";
    Dennis.flight.date.year = 2026;
    Dennis.flight.date.month = 1;
    Dennis.flight.date.day = 30;
    Dennis.flight.boarding_time.hour = "10";
    Dennis.flight.boarding_time.minute = "00";
    Dennis.flight.gate = "092";

    Dennis.send_email();
    return 0;
}
```

請完成名為 **tourist.h**與 **tourist.cpp** 的 C++ 程式，它們分別是 Tourist 與 Flight 類別的 **宣告定義** 與 **實作**

本題的相關程式將使用以下的 Makefile 進行編譯：

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

輸出相關程式碼模板可參考如下：

```
cout << "Dear " << ?????? << " " << ?????? << "," << endl;
cout << "Thank you for choosing Pineapple Airlines. We are excited to
welcome you on board for your upcoming flight to " << ?????? << ". Please
find your flight details below to ensure a smooth journey." << endl << endl;

cout << "Flight Information: " << endl;
cout << "\tFlight Number: " << ?????? << endl;
cout << "\tFrom: " << ?????? << endl;
cout << "\tTo: " << ?????? << endl;
cout << "\tDate: " << ?????? << "/" << ?????? << "/" << ?????? << endl;
cout << "\tBoarding Time: " << ?????? << ":" << ?????? << endl;
cout << "\tGate: " << ?????? << endl;
cout << "Safe travels,\nThe Pineapple Airlines Team" << endl;
```



先前提供給同學的程式碼中

```
cout << "\tBoarding Time: " << ?????? << ":" << ?????? << endl;
```

有拼寫錯誤，目前已更正。程式輸出時，接受以下兩種輸出：

- 「Boarding Time」...（此為正確拼寫）
- 「Borading Time」...（此為錯誤拼寫，助教手殘又打錯了）。

此題的執行結果可參考如下：

```
[3:23 user@ws p3] ./a.out
Dear Dennis Smith,
Thank you for choosing Pineapple Airlines. We are excited to welcome you on board for your upcoming flight to Hong Kong (HKG). Please find your flight details below to ensure a smooth journey.

Flight Information:
<TAB>Flight Number: CI 950
<TAB>From: Kaohsiung (KHH)
<TAB>To: Hong Kong (HKG)
<TAB>Date: 1/30/2026
<TAB>Boarding Time: 10:00 ←- 正確拼寫
<TAB>Gate: 092
Safe travels,
The Pineapple Airlines Team
[3:23 user@ws p3] ./a.out
Dear Dennis Smith,
Thank you for choosing Pineapple Airlines. We are excited to welcome you on board for your upcoming flight to Hong Kong (HKG). Please find your flight details below to ensure a smooth journey.

Flight Information:
<TAB>Flight Number: CI 950
<TAB>From: Kaohsiung (KHH)
<TAB>To: Hong Kong (HKG)
<TAB>Date: 1/30/2026
<TAB>Borading Time: 10:00 ←- 此處亦接受錯誤拼寫
<TAB>Gate: 092
Safe travels,
The Pineapple Airlines Team
[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。
- 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.cpp 與 Makefile 則不需繳交）：
  - tourist.h
  - tourist.cpp

*p4 學分計算機

Dennis 是屏西大學資訊工程學系的大二學生。屏西大學資訊工程學系的學分架構如下：

資工系	系必修	
	系選修	
通識教育	共同教育	
	體適能	
	博雅教育	美學與文化
		公民與社會
		自然與科技
國防教育（全民國防教育課程其學分不計入通識學分及各學系最低畢業學分，本題中需計入總學分的顯示）		
自由選修（在資工系選修、國防教育以及通識課程以外的其他所有課程）		

學期接近尾聲，獎勵預選課程的時間也快到了，Dennis 還有他的朋友們準備上校務系統選課。然而，這個學校的校務系統太難用了，該顯示的內容都沒有顯示，Dennis 根本不知道他的博雅教育（美學與文化、公民與社會與自然與科技）中的這三類各修了幾學分。因此 Dennis 想要開發一個可以告訴他每個類別各修習多少學分的程式。

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 "calculator.h"
using namespace std;

int main()
{
    CreditCalculator calculator;
    string line;
    while (getline(cin, line))
    {
        calculator.allocate_for_course(line); // 為每一個課程配置一個空間
    }
    calculator.show_score_info();
    calculator.credit_classify();
    calculator.free_space();
    calculator.credit.show_table();
    return 0;
}
```

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

```
#include <cstring>
#include <sstream>
#include <string>
#include <vector>
```

```
using namespace std;

class Course
{
public:
    string semester;     // 修課學期
    string subject_code; // 科目代號
    string subject_name; // 科目名稱
    unsigned credit;     // 該課程學分
    unsigned score;      // 取得分數

    void parse_str(string str);
};
```

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

```
#include "course.h"

void Course::parse_str(string str)
{
    istringstream iss(str);
    string token;
    vector<string> tokens;

 //使用 getline 分割字串, 遇到空白或 tab 分隔
    while (getline(iss, token, '\t'))
    {
        if (!token.empty())
        {
            tokens.push_back(token);
        }
    }

 //確保至少有基本欄位
    this->semester = tokens[0];

 //將 subject_code 和subject_name 分開處理
    istringstream subject_stream(tokens[1]);
    subject_stream >> this->subject_code;              // 提取 subject_code
    getline(subject_stream, this->subject_name);       // 剩下的作為
subject_name
    this->subject_name = this->subject_name.substr(1); // 去掉前導空格

 //檢查是否有學分欄位
    if (tokens.size() > 2 && !tokens[2].empty())
    {
        try
```

```
        {
            this->credit = stoi(tokens[2]);
        }
        catch (...)
        {
            this->credit = 0; // 若轉換失敗，預設為 0
        }
    }
    else
    {
        this->credit = 0;
    }

  // 檢查是否有分數或 "抵修"/"抵免" 欄位
    if (tokens.size() > 3 && !tokens[3].empty())
    {
        if ((tokens[3] == "抵修") || (tokens[3] == "抵免"))
        {
            this->score = 60; // 預設 60 分
        }
        else
        {
            try
            {
                this->score = stoi(tokens[3]);
            }
            catch (...)
            {
                this->score = 0; // 若轉換失敗，預設為 0
            }
        }
    }
    else
    {
        this->score = 0;
    }
}
```

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

```
#include "course.h"
#include <iostream>
#include <string>

class Credit
{
public:
    struct Csie
  {         // 資訊工程學系
```

```
        unsigned required = 0; // 必修課學分
        unsigned elective = 0; // 選修課
    } csie;

    struct GE
    {                                   // general education 通識教育
        unsigned common = 0;            // 共同教育
        unsigned physical_fitness = 0; // 體適能

        struct Liberal
        {                                     // liberal education 博雅教育
            unsigned aesthetics_culture = 0; // 美學文化
            unsigned civics_society = 0;     // 公民與社會
            unsigned science_technology = 0; // 自然與科技
        } liberal;
    } ge;

    unsigned free_elective = 0;         // 自由選修
    unsigned national_defence_edu = 0; // 國防教育

    void classify_and_add_credits(Course c); // 將每個課程獲得的學分填入到正確的
學分分類裡面
    void show_table();                       // 顯示學分分類後的表格
};
```

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

```
#include "credits.h"
#include "course_category.h"

void Credit::show_table()
{
    unsigned total_credit = csie.required + csie.elective + ge.common +
ge.physical_fitness +
                            ge.liberal.aesthetics_culture +
ge.liberal.civics_society +
                            ge.liberal.science_technology + free_elective +
national_defence_edu;

    cout << "==================================" << endl;
    cout << "資訊工程學系" << endl;
    cout << "\t系必修□" << csie.required << endl;
    cout << "\t系選修□" << csie.elective << endl;
    cout << "==================================" << endl;
    cout << "通識教育" << endl;
    cout << "\t共同教育□" << ge.common << endl;
    cout << "\t體適能□" << ge.physical_fitness << endl;
```

```
    cout << "\t博雅教育" << endl;
    cout << "\t\t> 美學與文化□" << ge.liberal.aesthetics_culture << endl;
    cout << "\t\t> 公民與社會□" << ge.liberal.civics_society << endl;
    cout << "\t\t> 自然與科技□" << ge.liberal.science_technology << endl;
    cout << "==================================" << endl;
    cout << "自由選修□" << free_elective << endl;
    cout << "國防教育□" << national_defence_edu << endl;
    cout << "==================================" << endl;
    cout << "總學分□" << total_credit << endl;
    cout << "==================================" << endl;
}

void Credit::classify_and_add_credits(Course c)
{
  // 請完成此函式
}
```

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

```
#include "credits.h"
#include <cstring>
#include <iostream>
#include <string>
#include <iomanip>

class CreditCalculator
{
public:
    Credit credit;
    Course *course = nullptr;
    unsigned course_counter = 0;

    void allocate_for_course(string line); // 為新的一個課程資料動態配置一個記憶體
空間
    void show_score_info();                // 輸出表格，檢查讀入的資訊是否有錯
    void credit_classify();                // 將取得的學分填入正確的分類
    void free_space();
};
```

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

```
#include "calculator.h"

void CreditCalculator::allocate_for_course(string line)
{
    Course *new_space = new Course[course_counter + 1];
```

```
    for (unsigned i = 0; i < course_counter; i++)
    {
        new_space[i] = course[i]; // 移動資料到新的空間
    }
    delete[] course;
    course = new_space;
    course[course_counter].parse_str(line); // 解析每一筆資訊
    course_counter++;
}

void CreditCalculator::show_score_info()
{
    cout << "[學期] [課程代號] [課程名稱]       [學分數] [成績]" << endl;
    for (int i = 0; i < course_counter; i++)
    {
        cout << left << setw(8)
             << course[i].semester << left << setw(10)
             << course[i].subject_code << left << setw(31)
             << course[i].subject_name << "\t"
             << course[i].credit << right << setw(10)
             << course[i].score << endl;
    }
}

void CreditCalculator::credit_classify()
{
    for (int i = 0; i < course_counter; i++)
    {
        credit.classify_and_add_credits(course[i]);
    }
}

void CreditCalculator::free_space(){
    delete [] course;
    course = nullptr;
}
```

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

```
#include <string>
using namespace std;
#define CSIE_REQUIRED_MAX 25              // 系必修修課程數量
#define CSIE_ELECTIVE_MAX 33              // 系選修修課程數量
#define GE_COMMON_MAX 22                  // 通識-共同教育
#define GE_PHYSICAL_FITNESS_MAX 24        // 通識-體適能
#define LIBERAL_CIVICS_SOCIETY_MAX 52     // 通識-博雅-公民與社會
```

```
#define LIBERAL_AESTHETICS_CULTURE_MAX 64 // 通識-博雅-美學與文化
#define LIBERAL_SCIENCE_TECHNOLOGY_MAX 41 // 通識-博雅-自然與科技
#define NATIONAL_DEFENCE_MAX 5            // 全民國防

string csie_required[] = {
  // 系必修
    "BFCZ002", // 程式設計一
    "BFCZ004", // 計算機概論
    "BFCZ005", // 線性代數
    "BFCZ170", // 微積分一
    "BFCZ171", // 微積分二
    "BFCZ006", // 程式設計二
    "BFCZ028", // 基本電學
    "BFCZ035", // 計算機網路
    "BFCZ013", // 資料結構
    "BFCZ017", // 計算機組織
    "BFCZ037", // 電子電路概論
    "BFCZ009", // 離散數學
    "BFCZ011", // 組合語言與微處理機
    "BFCZ036", // 資料庫系統導論
    "BFCZ040", // 物件導向程式設計
    "BFCZ012", // 系統程式
    "BFCZ015", // 機率與統計
..... 以下省略好多行
```

經過了一整天，Dennis 學分計算器終於完成八成了（所有輸入輸出全部都處理完了，目前的進度就剩下根據課程代號將已經及格的科目依照架構將學分分類，課程代號對應的框架請參考 course_category.h），但是不幸的是他剛到學校停車的時候碰到黑冠麻鷺突然掉進時空裂縫，短時間內恐怕是回不來了。他的朋友們從他的電腦中取得了尚未完成的程式碼，發現就剩下一個函式還沒有完成。假裝你是 Dennis 的朋友，你能夠協助完成最後一個尚未實作的函式嗎（此需實作的函式位於檔案 credits.cpp 中）。

請完成名為檔案 **credits.cpp** 中缺少函式實作的部分，其中包含 `Credit::classify_and_add_credits(Course c)` 函式的 **Implementation**：你需要先判斷引數傳入的 Course 物件 c 中的分數是否及格（c.score >= 60），若及格從 course_category.h 檔案提供的每個陣列尋找課程代號（c.subject_code）是否存在於

1. csie_required（系必修）
2. csie_elective（系選修）
3. ge_common（共同教育）
4. ge_physical_fitness（體適能）
5. liberal_civics_society（博雅教育：公民與社會）
6. liberal_aesthetics_culture（博雅教育：美學與文化）
7. liberal_science_technology（博雅教育：自然與科技）
8. national_defence（國防教育）

以上陣列中，若成功找到則將該課程學分（c.credit）加到 credits 對應的成員變數中；反之則將該課程學分計入「自由選修」學分中（this->free_elective）。

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp credits.o calculator.o course.o
    c++ main.cpp credits.o calculator.o course.o
calculator.o: calculator.cpp calculator.h
    c++ -c calculator.cpp
credits.o: credits.cpp credits.h credits.h course_category.h
    c++ -c credits.cpp
course.o: course.cpp course.h
    c++ -c course.cpp
clean:
    rm -f *.o *~ *.*~ a.out
```

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p4]▲./a.out▲<▲in.1↵
[學期]  [課程代號]  [課程名稱]                    [學分數]    [成績]
111/1▲▲▲GEC1212▲▲▲基礎學術英文(上)                2          60
111/1▲▲▲GEC1222▲▲▲基礎學術英文(下)                2          60
111/1▲▲▲GEC1111▲▲▲國
文(上)                                          2          60
111/1▲▲▲GEC4211▲▲▲體
育(上)                                          1          60
111/1▲▲▲GEC2211▲▲▲英語青少年文學                  2          60
111/1▲▲▲GEC2753▲▲▲細菌與人類疾病                  2          60
111/1▲▲▲BFCZ002▲▲▲程式設計(一)                   3          60
111/1▲▲▲BFCZ004▲▲▲計算機概
論                                              3          60
111/1▲▲▲BFCZ005▲▲▲線性代
數                                              3          60
111/1▲▲▲BFCZ170▲▲▲微積
分(一)                                          3          60
111/1▲▲▲BFCZ100▲▲▲網頁設計與應用                  3          60
111/1▲▲▲BFCZ136▲▲▲RFID概
論                                              3          60
111/2▲▲▲GEC1121▲▲▲國
文(下)                                          2          60
111/2▲▲▲GEC4221▲▲▲體
育(下)                                          1          60
111/2▲▲▲GEC2109▲▲▲生死學通
論                                              2          60
111/2▲▲▲BFCZ028▲▲▲基本電
學                                              3          60
111/2▲▲▲BFCZ006▲▲▲程式設計(二)                   3          60
111/2▲▲▲BFCZ035▲▲▲計算機網
路                                              3          60
111/2▲▲▲BFCZ171▲▲▲微積
分(二)                                          3          60
111/2▲▲▲BFCZ172▲▲▲數位系統導論暨實習              4          60
111/2▲▲▲BFCZ137▲▲▲Linux實務導論                  3          60
112/1▲▲▲GEC4202C▲▲體育-羽
```

```
球                                                2       60
112/1▲▲▲GEC2438▲▲▲全球化與兩岸關係                          2       60
112/1▲▲▲GEC2524▲▲▲統計分析與生活應用                        2       60
112/1▲▲▲BFCZ012▲▲▲系統程
式                                                3       60
112/1▲▲▲BFCZ013▲▲▲資料結
構                                                3       60
112/1▲▲▲BFCZ017▲▲▲計算機組
織                                                3       60
112/1▲▲▲BFCZ037▲▲▲電子電路概論                          3       60
112/1▲▲▲BFCZ038▲▲▲電子電路實習                          1       60
112/1▲▲▲BFCZ052▲▲▲視窗程式設計                          3       60
112/1▲▲▲BFCZ101▲▲▲科技英文(一)                          3       60
112/2▲▲▲GEC1233▲▲▲資訊英
文                                                2       60
112/2▲▲▲GEC2724▲▲▲創意學經
濟                                                2       60
112/2▲▲▲GEC2738▲▲▲當代應用心理學                          2       60
112/2▲▲▲BFCZ009▲▲▲離散數
學                                                3       60
112/2▲▲▲BFCZ011▲▲▲組合語言與微處理機                        3       60
112/2▲▲▲BFCZ036▲▲▲資料庫系統導論                          3       60
112/2▲▲▲BFCZ040▲▲▲物件導向程式設計                        3       60
112/2▲▲▲BFCZ104▲▲▲科技英文(二)                          3       60
==========↵
資訊工程學系
<TAB>系必修：53
<TAB>系選修：18
==========↵
通識教育
<TAB>共同教育：10
<TAB>體適能：4
<TAB>博雅教育
<TAB><TAB>>▲美學與文化：4
<TAB><TAB>>▲公民與社會：6
<TAB><TAB>>▲自然與科技：4
==========↵
自由選修：0
國防教育：0
==========↵
總學分：99
==========↵
[3:23▲user@ws▲p4]▲./a.out▲<▲in.2↵
[學期]  [課程代號]  [課程名稱]                                [學分數]     [成績]
111/1▲▲▲GEC1212▲▲▲基礎學術英文(上)                          2       60
111/1▲▲▲GEC1222▲▲▲基礎學術英文(下)                          2       60
111/1▲▲▲GEC1111▲▲▲國
文(上)                                           2       60
111/1▲▲▲GEC4211▲▲▲體
育(上)                                           1       60
111/1▲▲▲GEC2211▲▲▲英語青少年文學                          2       60
111/1▲▲▲GEC2753▲▲▲細菌與人類疾病                          2       60
111/1▲▲▲BFCZ002▲▲▲程式設計(一)                          3       60
```

111/1▲▲▲BFCZ004▲▲▲計算機概論 3 60
111/1▲▲▲BFCZ005▲▲▲線性代數 3 60
111/1▲▲▲BFCZ170▲▲▲微積分(一) 3 60
111/1▲▲▲BFCZ100▲▲▲網頁設計與應用 3 60
111/1▲▲▲BFCZ136▲▲▲RFID概論 3 60
111/2▲▲▲GEC1121▲▲▲國文(下) 2 60
111/2▲▲▲GEC4221▲▲▲體育(下) 1 60
111/2▲▲▲GEC2109▲▲▲生死學通論 2 60
111/2▲▲▲BFCZ028▲▲▲基本電學 3 60
111/2▲▲▲BFCZ006▲▲▲程式設計(二) 3 60
111/2▲▲▲BFCZ035▲▲▲計算機網路 3 60
111/2▲▲▲BFCZ171▲▲▲微積分(二) 3 60
111/2▲▲▲BFCZ172▲▲▲數位系統導論暨實習 4 60
111/2▲▲▲BFCZ137▲▲▲Linux實務導論 3 60
112/1▲▲▲GEC4202C▲▲體育-羽球 2 60
112/1▲▲▲GEC2438▲▲▲全球化與兩岸關係 2 60
112/1▲▲▲GEC2524▲▲▲統計分析與生活應用 2 60
112/1▲▲▲BFCZ012▲▲▲系統程式 3 60
112/1▲▲▲BFCZ013▲▲▲資料結構 3 60
112/1▲▲▲BFCZ017▲▲▲計算機組織 3 60
112/1▲▲▲BFCZ037▲▲▲電子電路概論 3 60
112/1▲▲▲BFCZ038▲▲▲電子電路實習 1 60
112/1▲▲▲BFCZ052▲▲▲視窗程式設計 3 60
112/1▲▲▲BFCZ101▲▲▲科技英文(一) 3 60
112/2▲▲▲GEC1233▲▲▲資訊英文 2 60
112/2▲▲▲GEC2724▲▲▲創意學經濟 2 60
112/2▲▲▲GEC2738▲▲▲當代應用心理學 2 60
112/2▲▲▲BFCZ009▲▲▲離散數學 3 60
112/2▲▲▲BFCZ011▲▲▲組合語言與微處理機 3 60
112/2▲▲▲BFCZ036▲▲▲資料庫系統導論 3 60
112/2▲▲▲BFCZ040▲▲▲物件導向程式設計 3 60
112/2▲▲▲BFCZ104▲▲▲科技英文(二) 3 60
113/1▲▲▲BFCZ015▲▲▲機率與統計 3 60
113/1▲▲▲BFCZ030▲▲▲微處理機實習 1 60

```
113/1▲▲▲BFCZ155▲▲▲軟體工
程                                3          60
113/1▲▲▲BFCZ059▲▲▲影像處理導論                    3          60
113/1▲▲▲BFCZ090▲▲▲資訊安
全                                3          60
113/1▲▲▲BFCZ166▲▲▲Android▲APP軟體設計(上)        3          60
113/1▲▲▲BFCZ177▲▲▲人工智慧程式設計                3          60
113/2▲▲▲BFCZ026▲▲▲作業系
統                                3          60
113/2▲▲▲BFCZ031▲▲▲實務專題(一)                    2          60
113/2▲▲▲BFCZ033▲▲▲演算
法                                3          60
113/2▲▲▲BFCZ055▲▲▲人工智慧導論                    3          60
113/2▲▲▲AXCZ132▲▲▲觀光日語(二)                    2          60
114/1▲▲▲BFCZ032▲▲▲實務專題(二)                    2          60
==========================================↵
資訊工程學系
<TAB>系必修：70
<TAB>系選修：33
==========================================↵
通識教育
<TAB>共同教育：10
<TAB>體適能：4
<TAB>博雅教育
<TAB><TAB>>▲美學與文化：4
<TAB><TAB>>▲公民與社會：6
<TAB><TAB>>▲自然與科技：4
==========================================↵
自由選修：2
國防教育：0
==========================================↵
總學分：133
==========================================↵
[3:23▲user@ws▲p4]▲
```



- 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。
- 複製所有檔案到當前位置（含兩個測試檔）

```
cp /home/stu/public/cpp2026s/cpp.hw7/p4/* .
```

- 本題提供的預設程式碼已可直接編譯與執行，唯輸出的學分表結果皆為 0。
- 完成本題後，同學們可以從校務系統中 [A081]_線上查詢 → [A0731S]_已修學分資料查詢」複製網頁中的資料製作製作自己的測試檔（不要複製表頭）。
- 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.cpp、Makefile 等檔案則不需繳交）：
  - credits.cpp

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: 334354

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=cpp:2026spring:hw7

Last update: **2026/05/04 10:36**