CPP Turnin作業5

國立屏東大學 資訊工程系 物件導向程式設計

Turnin作業5

- Turnin Code: cpp.hw5
- Due Date: 5/29 12:00 AM **Hard Deadline**

繳交方式說明

本次Turnin作業包含多個程式題，請為每一題建立一個資料夾，並將該題所要上傳的檔案放置其中後，再使用turnin指令上傳作業。請同學先為本次作業建立一個資料夾hw5，然後在hw5裡分別為每一題建立一個子資料夾，用以進行每一題的作答以及上傳。每一題的子資料夾名稱已寫於題目前方，請務必依照題目的規定建立子資料夾，例如第1題為p1，第2題為p2，餘依此類推。當我們完成某一個題目的作答後，就可以使用turnin指令將該題的答案上傳。以第1題為例，當我們在p1子資料夾裡完成作答後，就可以回到hw5資料夾，使用以下指令將其上傳：

[3:23 user@ws hw5] `turnin▴cpp.hw5▴p1↵`

當然，你也可以等到所有題目都完成後，回到hw5資料夾，使用以下指令將所有題目都加以上傳：

[3:23 user@ws hw5] `turnin▴cpp.hw5p▴.↵`

註：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。

---

p1 Date日期類別設計

請使用C++語言設計一個名為Date的類別，用以提供各式與日期相關的操作。Date類別將可以儲存一個日期的年、月與日三個短整數，並提供包含TW、US與UK三種區域格式：

- TW: 年/月/日
- US: MM/DD/YYYY
- UK: DD/MM/YYYY

舉例來說，以2024年1月21日為例，使用TW、US與UK格式可表示如下：

- TW: 113/1/21
- US: 1/21/2024
- UK: 21/1/2024

Date類別的物件，可以使用toString()函式取得代表該日期的字串(string類別的物件)，其內容係依據區域(TW、US與UK)與顯示格式(Short與Full)而定。以2024年1月21日為例，toString()傳回的字串內容如下：

- TW+Short: 113/1/21
- US+Short: 1/21/2024
- UK+Short: 21/1/2024
- TW+Full: 民國113年1月21日
- US+Short: January 21st, 2024
- UK+Short: 21st January, 2024

再以1911年5月2日為舉例如下：

- TW+Short: -1/5/2
- US+Short: 5/2/1911
- UK+Short: 2/5/1911
- TW+Full: 民國前1年5月2日
- US+Short: May 2nd, 1911
- UK+Short: 2nd May, 1911

注意：若是呼叫toString()時，沒有給予引數，那麼預設傳回US+Short格式的字串內容。

請參考以下的main.cpp程式：

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

int main()
{
    string dateStr;
    int m,d,y;

    Date d1 ("9/11/2021"); // 若無指定日期格式，預設為US
    Date d2;
    Date *d3, *d4=new Date("113/5/19", TW);
    Date d5;
    Date *older;

    cout << "Please input a date (US: MM/DD/YYYY): ";
    cin >> dateStr;
    d2.setDate(dateStr);  // 若無指定日期格式，預設為US
    cout << "Please input a date (UK: DD/MM/YYYY): ";
    cin >> dateStr;
    d3=new Date();
    d3->setDate(dateStr,UK);

    d4->setDate(*d3);

    d4->setYear(d4->getYear()-1);
    cout << "Month: ";
    cin >> m;
    d5.setMonth(m);
    cout << "Day: ";
```

```
    cin >> d;
    d5.setDay(d);
    cout << "Year: ";
    cin >> y;
    d5.setYear(y);

    cout << "d1:" << endl;
    cout << "Default: " << d1.toString() << endl;
    cout << "US: "      << d1.toString(US, Short) << endl;
    cout << "UK: "      << d1.toString(UK, Short) << endl;
    cout << "TW: "      << d1.toString(TW, Short) << endl;
    cout << "US-F: "    << d1.toString(US, Full) << endl;
    cout << "UK-F: "    << d1.toString(UK, Full) << endl;
    cout << "TW-F: "    << d1.toString(TW, Full) << endl;

    cout << "d2:" << endl;
    cout << "Default: " << d2.toString() << endl;
    cout << "US: "      << d2.toString(US, Short) << endl;
    cout << "UK: "      << d2.toString(UK, Short) << endl;
    cout << "TW: "      << d2.toString(TW, Short) << endl;
    cout << "US-F: "    << d2.toString(US, Full) << endl;
    cout << "UK-F: "    << d2.toString(UK, Full) << endl;
    cout << "TW-F: "    << d2.toString(TW, Full) << endl;
    cout << "d3:" << endl;
    cout << "Default: " << d3->toString() << endl;
    cout << "US: "      << d3->toString(US, Short) << endl;
    cout << "UK: "      << d3->toString(UK, Short) << endl;
    cout << "TW: "      << d3->toString(TW, Short) << endl;
    cout << "US-F: "    << d3->toString(US, Full) << endl;
    cout << "UK-F: "    << d3->toString(UK, Full) << endl;
    cout << "TW-F: "    << d3->toString(TW, Full) << endl;

    cout << "d4:" << endl;
    cout << "Default: " << d4->toString() << endl;
    cout << "US: "      << d4->toString(US, Short) << endl;
    cout << "UK: "      << d4->toString(UK, Short) << endl;
    cout << "TW: "      << d4->toString(TW, Short) << endl;
    cout << "US-F: "    << d4->toString(US, Full) << endl;
    cout << "UK-F: "    << d4->toString(UK, Full) << endl;
    cout << "TW-F: "    << d4->toString(TW, Full) << endl;
    cout << "d5:" << endl;
    cout << "Default: " << d5.toString() << endl;
    cout << "US: "      << d5.toString(US, Short) << endl;
    cout << "UK: "      << d5.toString(UK, Short) << endl;
    cout << "TW: "      << d5.toString(TW, Short) << endl;
    cout << "US-F: "    << d5.toString(US, Full) << endl;
    cout << "UK-F: "    << d5.toString(UK, Full) << endl;
    cout << "TW-F: "    << d5.toString(TW, Full) << endl;
```

```
    cout << endl;

    if(d2.compare(*d3)==0)
    {
        cout << "d2 is equal to d3.\n";
    }
    else
    {
        older=d3->older(&d2);
        cout << "The date " << older->toString() << " is earlier than ";
        if(d3->compare(d2)>0)
            cout << d3->toString() << ".\n";
        else
            cout << d2.toString() << ".\n";
    }
}
```

關於Date類別的定義，請參考以下的date.h檔案：

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

enum Locale {TW, US, UK};
enum DateFormat { Short, Full};

class Date
{
private:
    short int year;
    short int month;
    short int day;
public:
    Date();
    Date(string dateStr);
    Date(string dateStr, Locale loc);
    void setDate(string dateStr);
    void setDate(string dateStr, Locale loc);
    void setDate(Date d);

    void setYear(short int y);
    short int getYear();
    void setMonth(short int m);
    short int getMonth();
    void setDay(short int d);
    short int getDay();
    Date *older(Date *d);
    int compare(Date d);
    string toString();
    string toString(Locale loc, DateFormat formt);
```

```
};
```

你必須完成名為date.cpp的C++語言程式，其中包含Date類別的實作。此程式完成後，可讓使用者輸入兩個日期，並將宣告在main.cpp裡的d1、d2、d3、d4、d5等日期輸出，並判定d3與d2的關係。此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please␣input␣a␣date␣(US:␣MM/DD/YYYY):␣10/01/1998↵
Please␣input␣a␣date␣(UK:␣DD/MM/YYYY):␣30/9/1998↵
Month:␣1↵
Day:␣01↵
Year:␣1900↵
d1:↵
Default:␣9/11/2021↵
US:␣9/11/2021↵
UK:␣11/9/2021↵
TW:␣110/9/11↵
US-F:␣September␣11th,␣2021↵
UK-F:␣11th␣September,␣2021↵
TW-F:␣民國110年9月11日
d2:↵
Default:␣10/1/1998↵
US:␣10/1/1998↵
UK:␣1/10/1998↵
TW:␣87/10/1↵
US-F:␣October␣1st,␣1998↵
UK-F:␣1st␣October,␣1998↵
TW-F:␣民國87年10月1日
d3:↵
Default:␣9/30/1998↵
US:␣9/30/1998↵
UK:␣30/9/1998↵
TW:␣87/9/30↵
US-F:␣September 30th, 1998↵
UK-F:␣30th September, 1998↵
TW-F:␣民國87年9月30日
d4:↵
Default:␣9/30/1997↵
US:␣9/30/1997↵
UK:␣30/9/1997↵
TW:␣86/9/30↵
US-F:␣September␣30th,␣1997↵
UK-F:␣30th␣September,␣1997↵
TW-F:␣民國86年9月30日
d5:↵
Default:␣1/1/1900↵
US:␣1/1/1900↵
```

```
UK:▲1/1/1900↵
TW:▲-12/1/1↵
US-F:▲January 1st,▲1900↵
UK-F:▲1st▲January,▲1900↵
TW-F:▲民國前12年1月1日
↵
The▲date▲9/30/1998▲is▲earlier▲than▲10/1/1998.↵
[3:23 user@ws hw]
```

本題的相關程式將使用以下的Makefile進行編譯：

```
all: main.cpp date.o
    c++ main.cpp date.o

date.o: date.cpp date.h
    c++ -c date.cpp

clean:
    rm -f *.o *~ *.*~ a.out
```

請注意本題只需繳交date.cpp□其它檔案則不需繳交。

提示：你可能會需要使用定義在sstream裡的stringstream類別，請自行查閱相關資料。

p2 明星Star類別

請使用C++語言設計一個名為Star的類別，用以提供明星的基本資料維護□Star類別將可以儲存一個明星的姓名、性別、生日與地址，請先參考以下的star.h所定義的類別內容：

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

enum Gender
{
    Male,
    Female,
    Undef
};

class Star
{
private:
    string name;
```

```
    Gender gender;
    Date birthday;
    char *address;

public:
    Star();
    Star(string name);
    Star(string name, Gender gender);
    Star(string name, Gender gender, string birthday, string address);
    Star(string name, Gender gender, string address);

    void setName(string n);
    string getName();
    void setGender(Gender g);
    Gender getGender();
    void setBirthday(string s);
    void setBirthday(Date d);
    void setBirthday(string s, Locale l);
    Date getBirthday();
    void setAddress(string s);
    string getAddress();
    void setAStar(string name, Gender gender, string birthday, string
address);
    void show()
    {
        cout << name << "("
             << (gender == Male ? "Male" : gender == Female ? "Female" :
"Undef")
             << "," << birthday.toString() << ") Address: " << address <<
endl;
    }
};
```

注意:在Star類別裡的birthday是使用p1所設計的Date類別進行宣告。

請完成star.cpp的程式碼，將上述定義在star.h裡的Star類別的建構函式與各個操作方法(member functions)加以實作完成，並使用下列main.cpp進行測試：

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

int main()
{
  Star tom;
```

```
  tom.setName("Tom Cruise");
  tom.setGender(Male);
  tom.setBirthday("7/31/1962");
  tom.setAddress("No. 19, Happy Rd, NY");

  Star *kelly = new Star("Kelluy McGillis");
  kelly->setBirthday(Date("9/7/1957", UK));
  kelly->setGender(Female);

  Star val("Val Edward Kilmer", Male, "12/31/1959", "19 La Lita Ln, Santa
Barbara, CA 93105");
  Star meg("Meg Ryan", Female, "No. 19, Happy Rd, NY");
  meg.setBirthday(Date("12/19/1961"));

  tom.show();
  kelly->show();
  val.show();
  meg.show();
}
```

請參考以下的Makefile

```
all: main.cpp star.o date.o
    c++ main.cpp star.o date.o
star.o: star.cpp star.h
    c++ -c star.cpp

date.o: date.cpp date.cpp
    c++ -c date.cpp

clean:
    rm -f *.o *~ *.*~ a.out
```

本題的執行結果如下：

```
 [3:23 user@ws hw] ./a.out↵
 Tom▲Cruise(Male,7/31/1962)▲Address:▲No.▲19,▲Happy▲Rd,▲NY↵
 Kelluy▲McGillis(Female,7/9/1957)▲Address:▲Undef↵
 Val▲Edward▲Kilmer(Male,12/31/1959)▲Address:▲19▲La▲Lita▲Ln,▲Santa▲Barbara,▲CA↵
 Meg▲Ryan(Female,12/19/1961)▲Address:▲No.▲101,▲Almond▲St,▲PA↵
 [3:23 user@ws hw]
```

請特別注意，此題Star類別中的address被宣告為char *請依據其地址的字數配置動態地配置適切的記憶體位置（例如“No.▲19,▲Happy▲Rd,▲NY”應使用new char[21]進行配置。此部份將會由助教進行人工的檢查，若沒有正確地進行動態記憶體配置，此題不予計分。

請注意本題只需繳交star.cpp以及date.cpp□其它檔案則不需繳交。

p3 明星管理

承上題。當你完成了Star類別的設計與實作後，現在可以進一步寫一個簡單的明星管理程式，讓我們可以輸入明星的資料。此程式預設先配置5個Star類別的物件所需的記憶體空間，並可在程式執行的過程中，依使用者指令將記憶體空間變為原本的兩倍。

此題程式執行時，將接收以下指令：

- i: 新增一個明星的資料
- l: 列出所有明星資料
- d: 將既有的記憶體空間倍增
- q: 結束程式

此題的執行結果可參考如下（日期皆為US美式）：

```
[3:23 user@ws hw] ./a.out↵
<CMD>?▲l↵
#1:▲—▲empty▲—↵
#2:▲—▲empty▲—↵
#3:▲—▲empty▲—↵
#4:▲—▲empty▲—↵
#5:▲—▲empty▲—↵
<CMD>?▲i↵
Name:▲Tom▲Cruise↵
Gender▲(F/M):▲M↵
Birthday▲(MM/DD/YYYY):▲7/31/1962↵
Address:▲No.▲19,▲Happy▲Rd.,▲NY↵
<CMD>?▲i↵
Name:▲Kelluy▲McGillis ↵
Gender▲(F/M):▲F↵
Birthday▲(MM/DD/YYYY):▲7/9/1957↵
Address:▲I▲don't▲know↵
<CMD>?▲i↵
Name:▲Val▲Edward▲Kilmer↵
Gender▲(F/M):▲M↵
Birthday▲(MM/DD/YYYY):▲12/31/1959↵
Address:▲19▲La▲Lita▲Ln,▲Santa▲Barbara,▲CA↵
<CMD>?▲i↵
Name:▲Meg▲Ryan↵
Gender▲(F/M):▲F↵
Birthday▲(MM/DD/YYYY):▲12/19/1961↵
Address:▲No.▲101,▲Almond▲St,▲PA↵
<CMD>?▲l↵
#1:▲Tom▲Cruise(Male,7/31/1962)▲Address:▲No.▲19,▲Happy▲Rd.,▲NY↵
#2:▲Kelluy▲McGillis(Female,7/9/1957)▲Address:▲I▲don't▲known↵
```

```
#3:▲Val▲Edward▲Kilmer(Male,12/31/1959)▲Address:▲19▲La▲Lita▲Ln,▲Santa▲Barbara,▲CA↵
#4:▲Meg▲Ryan(Female,12/19/1961)▲Address:▲No.▲101,▲Almond▲St,▲PA↵
#5:▲—▲empty▲—↵
<CMD>?▲i↵
Name:▲Anthony▲Edwards↵
Gender▲(F/M):▲M↵
Birthday▲(MM/DD/YYYY):▲7/19/1962↵
Address:▲300▲N▲Los▲Carneros▲Rd,▲Goleta,▲CA↵
<CMD>?▲i↵
Not▲enough▲space!↵
<CMD>?▲l↵
#1:▲Tom▲Cruise(Male,7/31/1962)▲Address:▲No.▲19,▲Happy▲Rd.,▲NY↵
#2:▲Kelluy▲McGillis(Female,7/9/1957)▲Address:▲l▲don't▲known↵
#3:▲Val▲Edward▲Kilmer(Male,12/31/1959)▲Address:▲19▲La▲Lita▲Ln,▲Santa▲Barbara,▲CA↵
#4:▲Meg▲Ryan(Female,12/19/1961)▲Address:▲No.▲101,▲Almond▲St,▲PA↵
#5:▲Anthony▲Edwards(Male,7/19/1962)▲Address:▲300▲N▲Los▲Carneros▲Rd,▲Goleta,▲CA↵
<CMD>?▲d↵
Space▲doubled!↵
<CMD>?▲l↵
#1:▲Tom▲Cruise(Male,7/31/1962)▲Address:▲No.▲19,▲Happy▲Rd.,▲NY↵
#2:▲Kelluy▲McGillis(Female,7/9/1957)▲Address:▲l▲don't▲known↵
#3:▲Val▲Edward▲Kilmer(Male,12/31/1959)▲Address:▲19▲La▲Lita▲Ln,▲Santa▲Barbara,▲CA↵
#4:▲Meg▲Ryan(Female,12/19/1961)▲Address:▲No.▲101,▲Almond▲St,▲PA↵
#5:▲Anthony▲Edwards(Male,7/19/1962)▲Address:▲300▲N▲Los▲Carneros▲Rd,▲Goleta,▲CA↵
#6:▲—▲empty▲—↵
#7:▲—▲empty▲—↵
#8:▲—▲empty▲—↵
#9:▲—▲empty▲—↵
#10:▲—▲empty▲—↵
<CMD>?▲i↵
Name:▲Thomas▲Skerritt↵
Gender▲(F/M):▲M↵
Birthday▲(MM/DD/YYYY):▲8/25/1933↵
Address:▲10125▲E▲Jefferson▲Ave,▲Detroit,▲MI↵
<CMD>?▲l↵
#1:▲Tom▲Cruise(Male,7/31/1962)▲Address:▲No.▲19,▲Happy▲Rd.,▲NY↵
#2:▲Kelluy▲McGillis(Female,7/9/1957)▲Address:▲l▲don't▲known↵
#3:▲Val▲Edward▲Kilmer(Male,12/31/1959)▲Address:▲19▲La▲Lita▲Ln,▲Santa▲Barbara,▲CA↵
#4:▲Meg▲Ryan(Female,12/19/1961)▲Address:▲No.▲101,▲Almond▲St,▲PA↵
#5:▲Anthony▲Edwards(Male,7/19/1962)▲Address:▲300▲N▲Los▲Carneros▲Rd,▲Goleta,▲CA↵
#6:▲Thomas▲Skerritt(Male,8/25/1933)▲Address:▲10125▲E▲Jefferson▲Ave,▲Detroit,▲MI↵
#7:▲—▲empty▲—↵
#8:▲—▲empty▲—↵
#9:▲—▲empty▲—↵
#10:▲—▲empty▲—↵
<CMD>?▲q↵
[3:23 user@ws hw]
```

本題的相關程式將使用以下的Makefile進行編譯：

```
main: main.cpp star.o date.o
    c++ main.cpp star.o date.o

star.o: star.cpp star.h
    c++ -c star.cpp

date.o: date.cpp date.cpp
    c++ -c date.cpp

clean:
    rm -f *.o *~ *.*~ a.out
```

請注意本題只需繳交main.cpp、star.cpp與date.cpp，其它檔案則不需繳交。

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: 281773

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2024spring-draft

Last update: **2024/05/21 16:54**