Simple class creation

Introduction to Object Oriented programming

Requirements - Project MockData

Generate **30** students into **mockData.txt** file

Each student should have the information of

+ Student’s ID
+ Student’s fullname
+ Address
+ Email
+ Telephone number
+ Date of birth
+ GPA (grade point average)

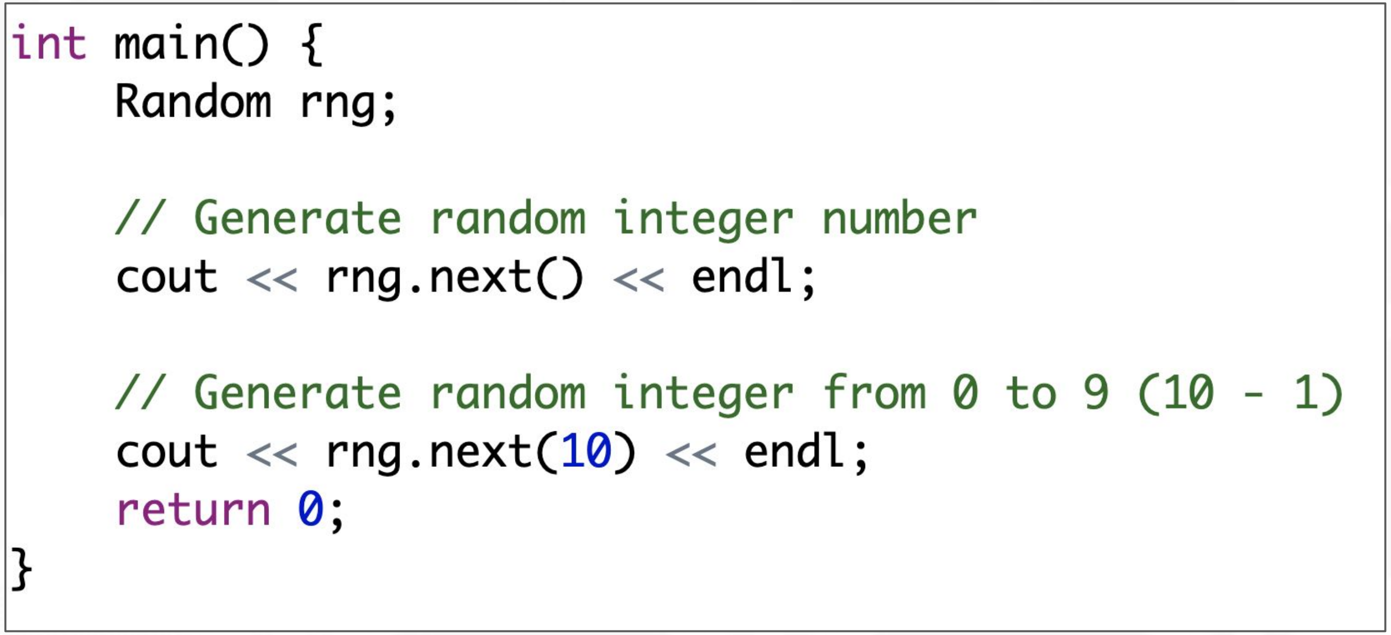
HINTS

Example Random number

```
#include <stdlib.h>

// Khởi tạo bộ sinh số ngẫu nhiên

srand(time(NULL));


// Sinh số nguyên ngẫu nhiên

cout << rand();
```

Random class

Implement a class for generating random number named **RandomIntegerGenerator**



```
int main() {
    Random rng;

    // Generate random integer number
    cout << rng.next() << endl;

    // Generate random integer from 0 to 9 (10 - 1)
    cout << rng.next(10) << endl;
    return 0;
}
```

VnNameGenerator

- ❏ Choose first name from this list: https://vi.wikipedia.org/wiki/H%E1%BB%8D_ng%C6%B0%E1%BB%9Di_Vi%E1%BB%87t_Nam

- ❏ Choose middle name from this list: http://www.erct.com/4-ChiaSe/SuuTam/Tinh_danh-TEN_DEM.htm

- ❏ Choose last name from this list https://xltiengviet.fandom.com/wiki/T%C3%AAn_ng%C6%B0%E1%BB%9Di_Vi%E1%BB%87t_Nam

In main function, generate 20 fake names.

Hint

Fullname is the entity class for **storing** data. It should have 3 attributes: *_firstName, _middleName, _lastName*

VnNameGenerator is the **business** class, for **generating data** using next function

Fullname **VnNameGenerator** ::next()

Fake Address

HcmAddressGenerator

1. Goto tiki.vn
2. Find list of district (choose 5)
3. Find list of ward for each district (choose 8 for each district)
4. Find list of street from each district (choose 5 - Google maps)
5. Create a combination from these elements

Hint

Address is the **entity** class for **storing** data. It should has 4 attributes:

_number: The number of the house (should be string because of something like this 22/34, 6 bis…). You may need a class named FakeHouseNumber which acts as a business class for generating house number.

_street: The name of the street.

_ward: string

_district: string

_city: string.

HcmAddressGenerator is the **business** class for **generating** data

Address **HcmAddressGenerator**::next()

Fake Tel

VnTelNumberGenerator

Pick an operator from this list and generate the rest:

https://quantrimang.com/danh-sach-dau-so-cac-mang-di-dong-o-viet-nam-133203

Note: Display of tel is different like 0909 222 888

Hints

The telephone number could be a string.

VnTelNumberGenerator is the business class

    string VnTelNumberGenerator::next()

Fake Email

EmailGenerator

Prepare **10** biggest company domains like gmail.com, microsoft.com, apple.com, amazon.com....

1. Generate fake fullname, for example Tran Duy Quang
2. Then choose one company domain, like apple.com
3. Then generate fake email like tdquang@apple.com

(Hint: get substring, first letter of Firstname and Middle name)

Hints

Email could be a string

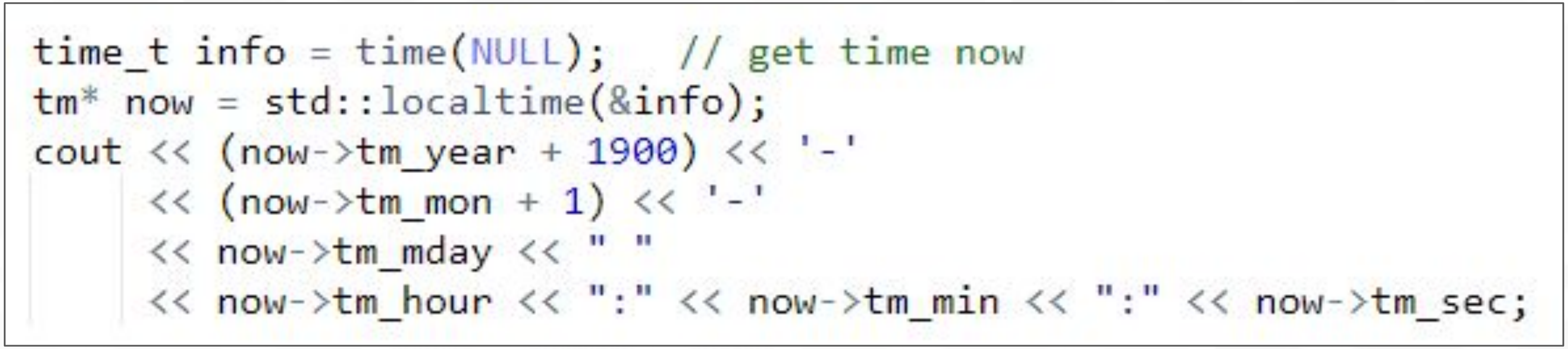
EmailGenerator is a business class.

```
    string EmailGenerator::next()
```

Time

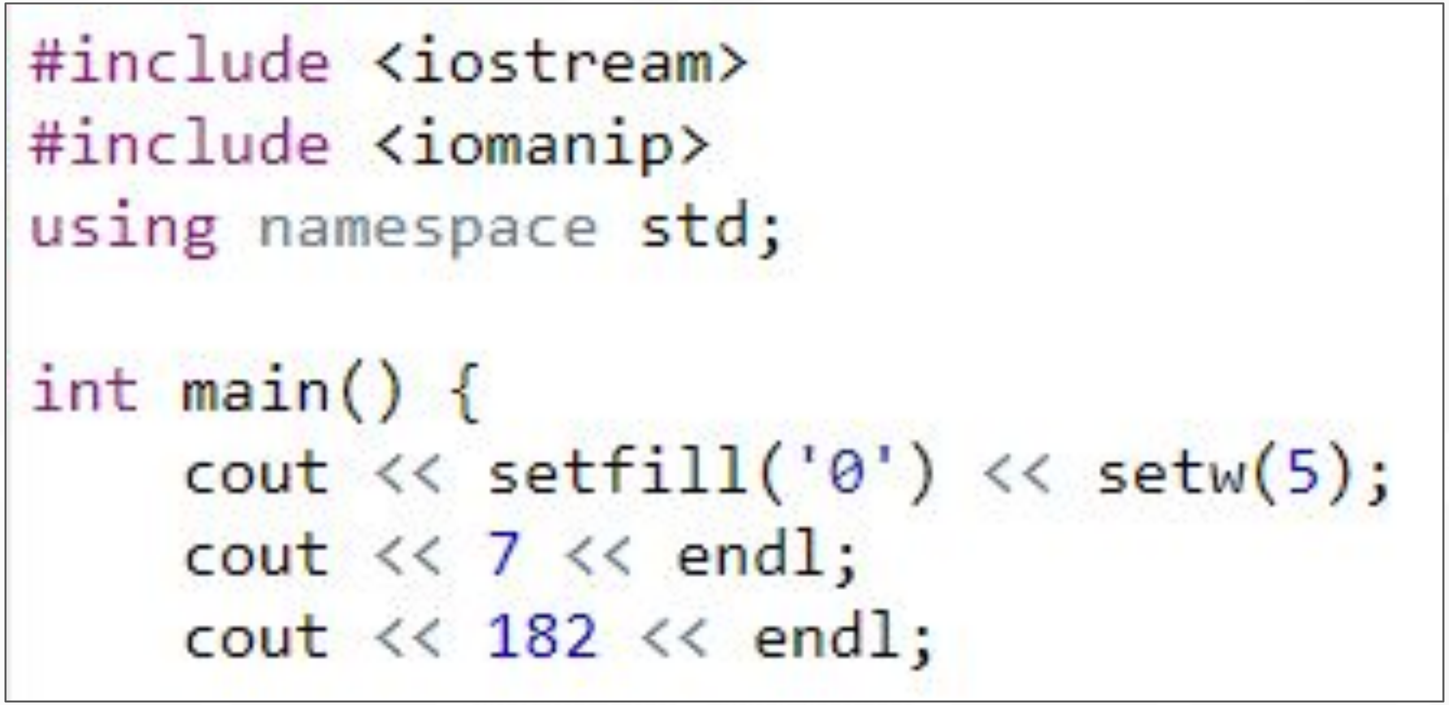
Example Get current time

#include <ctime>



```
time_t info = time(NULL);   // get time now
tm* now = std::localtime(&info);
cout << (now->tm_year + 1900) << '-'
     << (now->tm_mon + 1) << '-'
     << now->tm_mday << " "
     << now->tm_hour << ":" << now->tm_min << ":" << now->tm_sec;
```

Example Set width of output



```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    cout << setfill('0') << setw(5);
    cout << 7 << endl;
    cout << 182 << endl;
```

Output:

```
00007
182
```

Project Time

1. Time(): initialize with current time
2. Time(int, int, int): initialize using 3 components hour, minute, second
3. string toString(): output in this format “06:18:20”
4. Time Time::parse(string): convert a string like “06:18:20” into time
5. bool Time::tryParse(string, &Time)
6. bool Time::isValid(string)

Date

11. Project Date

1. Date(): initialize with current date
2. Date(int, int, int): initialize using 3 components hour, minute, second
3. string toString(): output in this format “07/06/2020”
4. Date Date::parse(string): convert a string like “07/06/2020” into time
5. bool Date::tryParse(string, &Date)
6. bool Date::isValid(string)
7. bool Date::isLeapYear(int)

Requirement

In main function, write code to test all of your class methods

What adjustment would you make to output date in short format 06/07/2020 or in long format 06/07/2020?

What if your app is used in US, where 07/06/2020 is the correct format?

Fake Birthday

BirthdayGenerator

Birthday is just a date with some constraints

As of 11 November 2019, the oldest known living person is Kane Tanaka of Japan, aged 116 years, 313 days. (https://en.wikipedia.org/wiki/List_of_the_verified_oldest_people#:~:targetText=The%20oldest%20person%20ever%20whose,of%20116%20years%2C%2054%20days)

Generate birthday using function next

```
Date BirthdayGenerator::next()
```

BirthdayGenerator enhancement

Date **BirthdayGenerator**::next(int age)

Generate a random birthday with a specific age (which means you only have to generate random day and month only, the year can be inferred from age and current year)