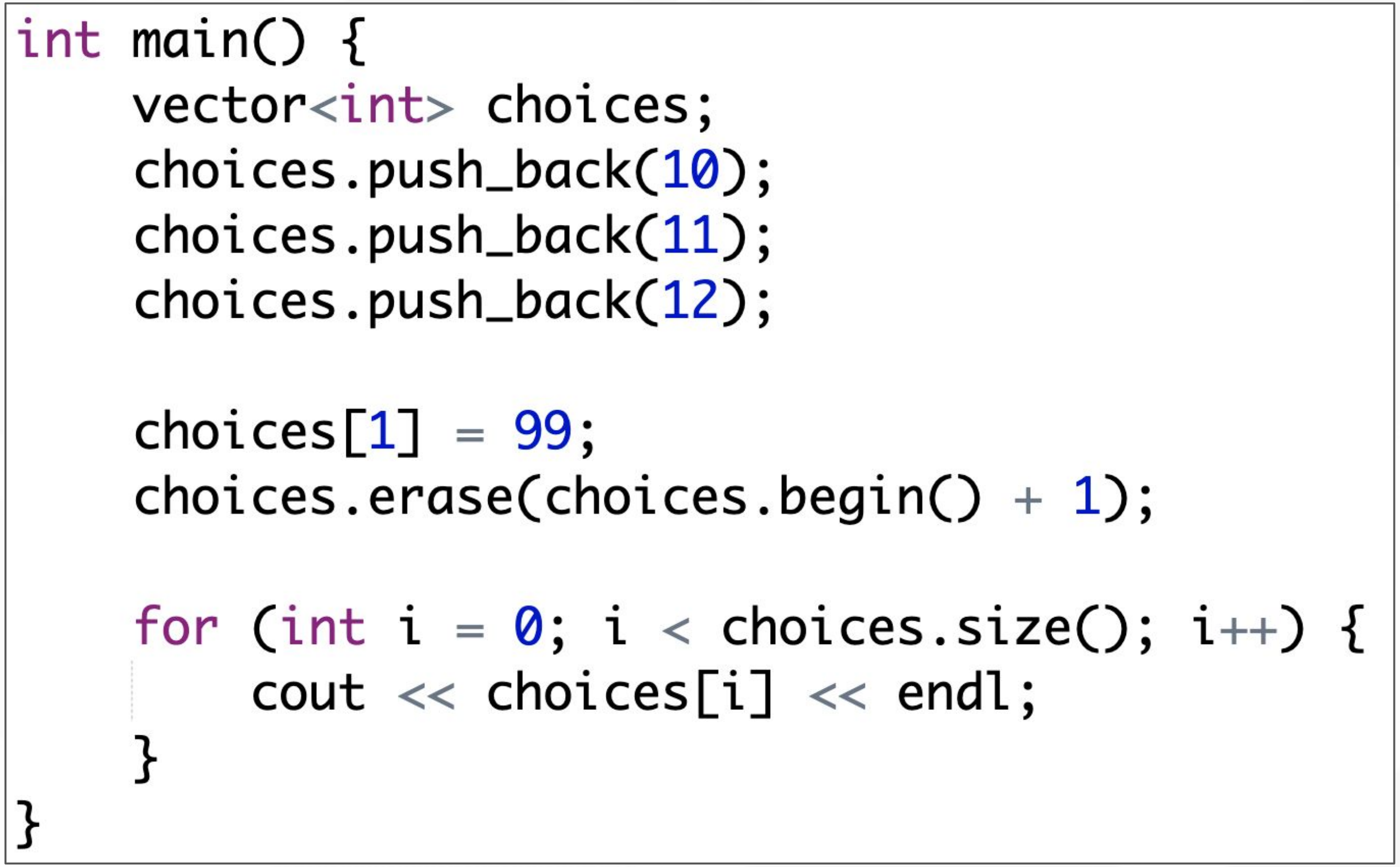
Simple class creation

Introduction to Object Oriented programming

Example: Working with vector (Dynamic array)



```
int main() {
    vector<int> choices;
    choices.push_back(10);
    choices.push_back(11);
    choices.push_back(12);

    choices[1] = 99;
    choices.erase(choices.begin() + 1);

    for (int i = 0; i < choices.size(); i++) {
        cout << choices[i] << endl;
    }
}
```

General guidelines

- ❑ Each class should be split into 2 files
  - ➢ **ClassName**.h (declaration)
  - ➢ **ClassName**.cpp (implementation)
- ❑ For example, suppose we have a class named **Student**
- ❑ We should have 2 files
  - ➢ **Student**.h for declaration
  - ➢ **Student**.cpp for implementation

Random

Example Random number

```
#include <stdlib.h>

// Khởi tạo bộ sinh số ngẫu nhiên

srand(time(NULL));


// Sinh số nguyên ngẫu nhiên

cout << rand();
```

01. Project Random

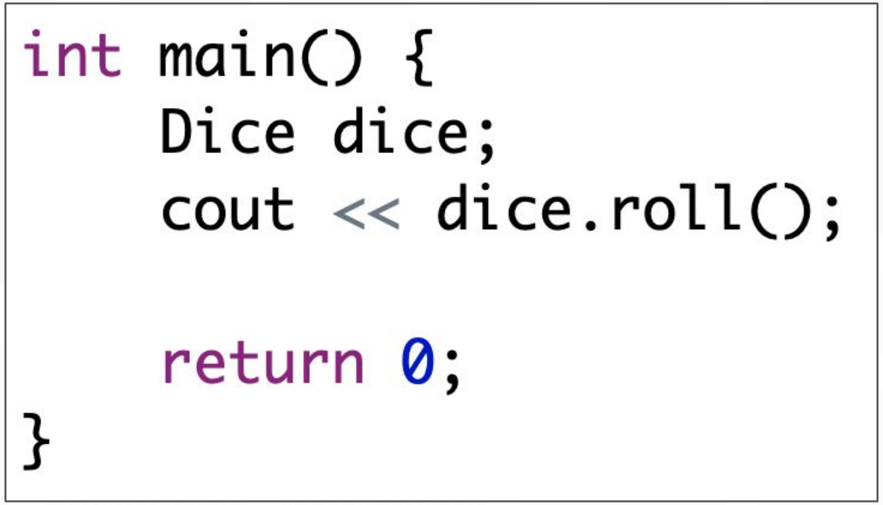
Implement a class for generating random number named **Random**

```cpp
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;
}
```

Dice

02. Project Dice

Create a class that simulates a dice



```
int main() {
    Dice dice;
    cout << dice.roll();

    return 0;
}
```

Dice’s enhancement

1. How to know the number of times a dice has been rolled?

    Roll infinitely until enter 0

1. What is the average number that we get from all the rolls?
2. Print out all the counts for all the values of one dice

    For example

    1: 6 times, 2: 4 times, 3: 0 times, 4: 7 times….

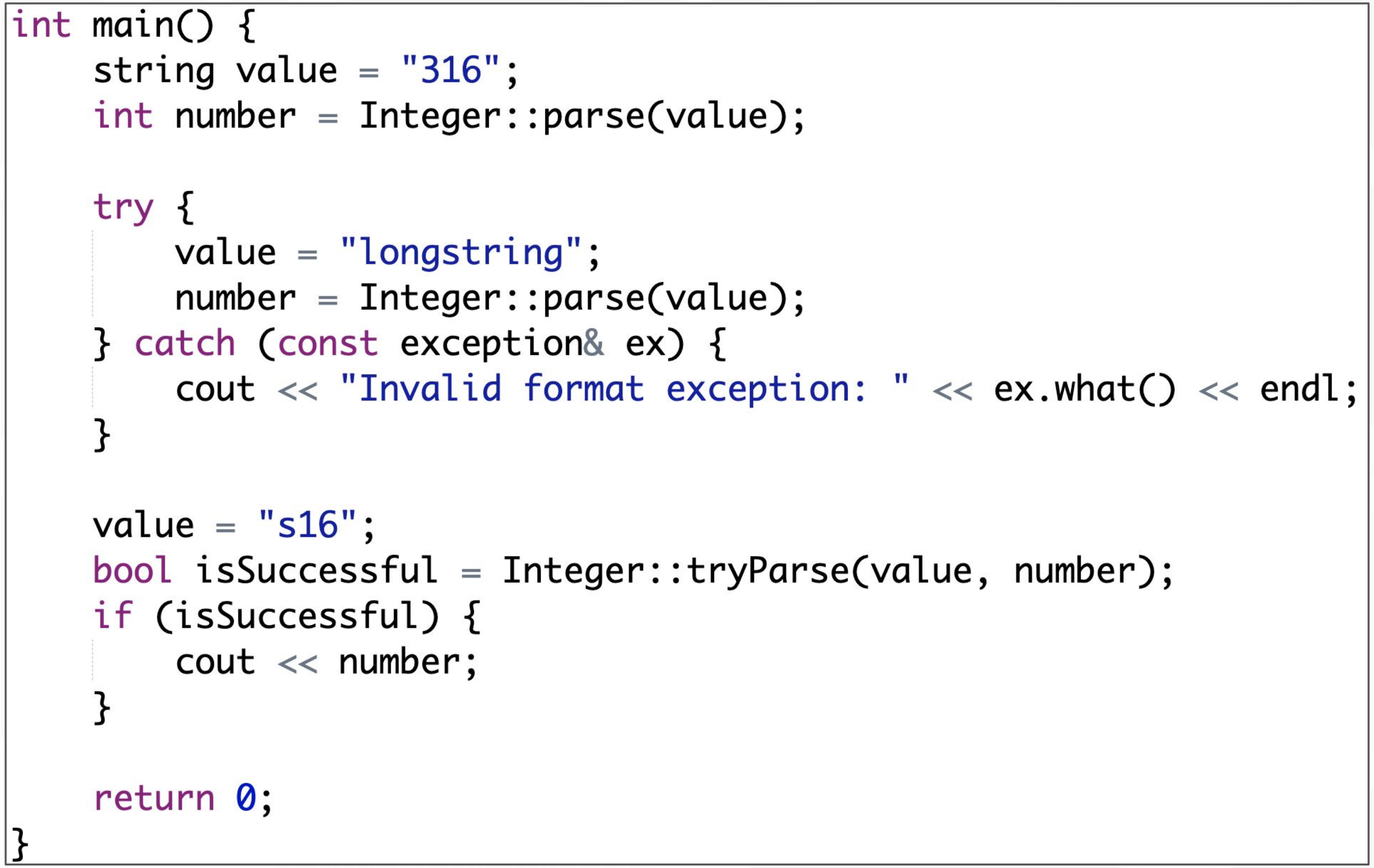
    _counts[7] = [-1, 0, 0, 0, 0, 0, 0]   _count[1]++

Parsing string into integer

Example: Parsing string into number

```
string value = “12”;

int number1  = stoi(value);

int number2 = stof(value);
```

03. Project Integer



```cpp
int main() {
    string value = "316";
    int number = Integer::parse(value);

    try {
        value = "longstring";
        number = Integer::parse(value);
    } catch (const exception& ex) {
        cout << "Invalid format exception: " << ex.what() << endl;
    }

    value = "s16";
    bool isSuccessful = Integer::tryParse(value, number);
    if (isSuccessful) {
        cout << number;
    }

    return 0;
}
```

Integer’s enhancements (1)

1. **Check if an integer is a prime number**

   Integer::isPrime(int)

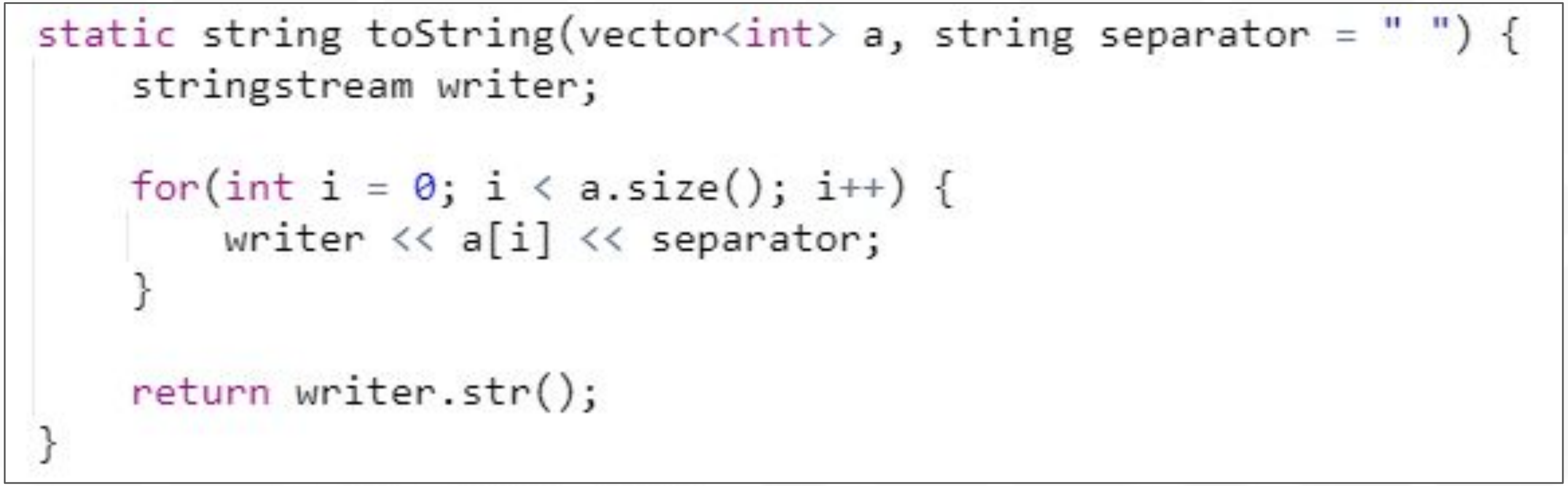
1. **Check if an integer is a perfect number** (sum of all n’s divisors = 2*n) Integer::isPerfect(int)

   For example: **6** has 4 divisors [1, 2, 3, 6],

   1 + 2 + 3 + 6 = 2 ***6** ==> **6** is a perfect number

Integer’s enhancements (2)

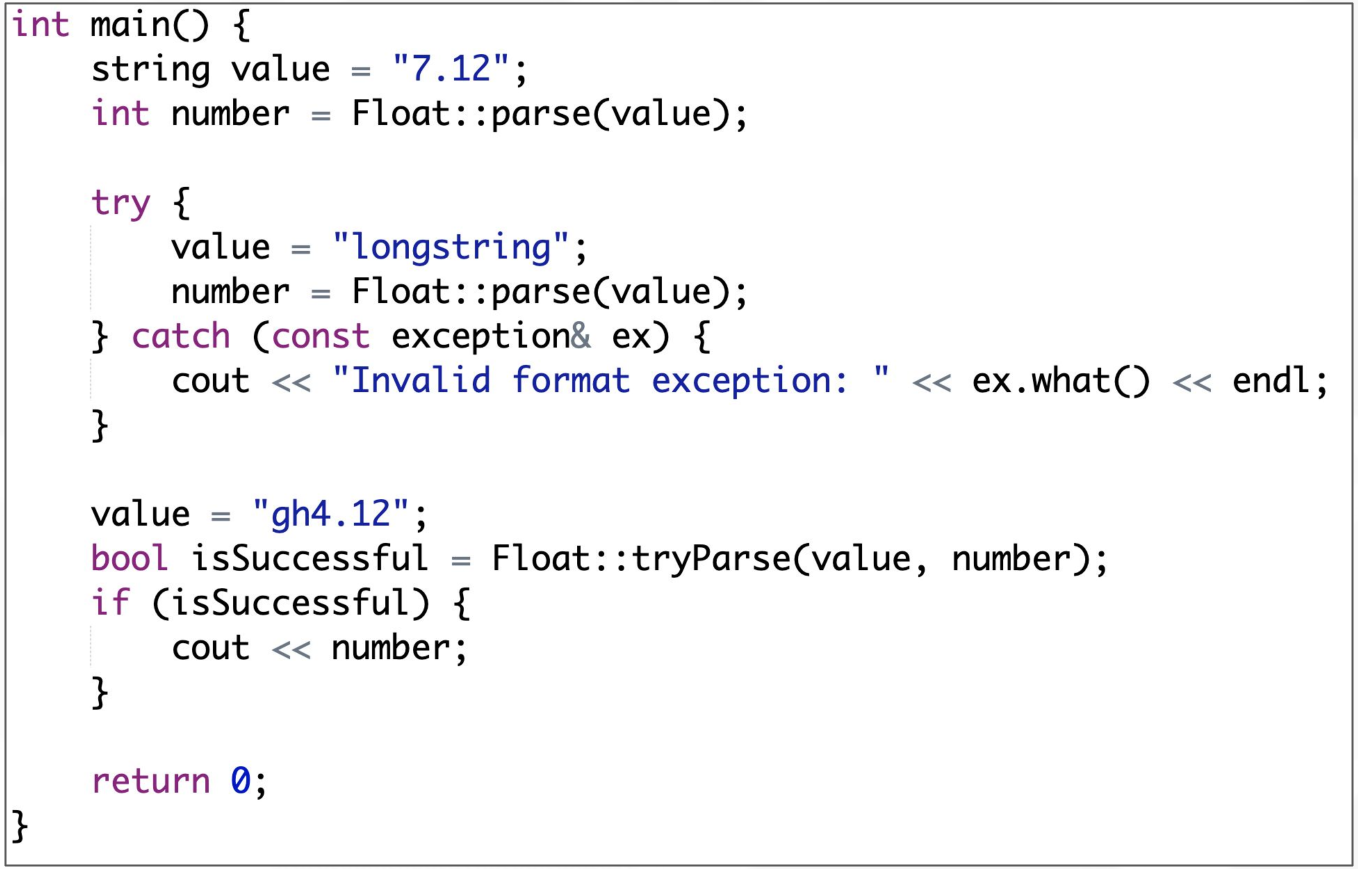
3. Convert a vector of int into string (only do this if already introduced vector)



```
static string toString(vector<int> a, string separator = " ") {
    stringstream writer;

    for(int i = 0; i < a.size(); i++) {
        writer << a[i] << separator;
    }

    return writer.str();
}
```

Parsing string into float

04. Project Float



```cpp
int main() {
    string value = "7.12";
    int number = Float::parse(value);

    try {
        value = "longstring";
        number = Float::parse(value);
    } catch (const exception& ex) {
        cout << "Invalid format exception: " << ex.what() << endl;
    }

    value = "gh4.12";
    bool isSuccessful = Float::tryParse(value, number);
    if (isSuccessful) {
        cout << number;
    }

    return 0;
}
```

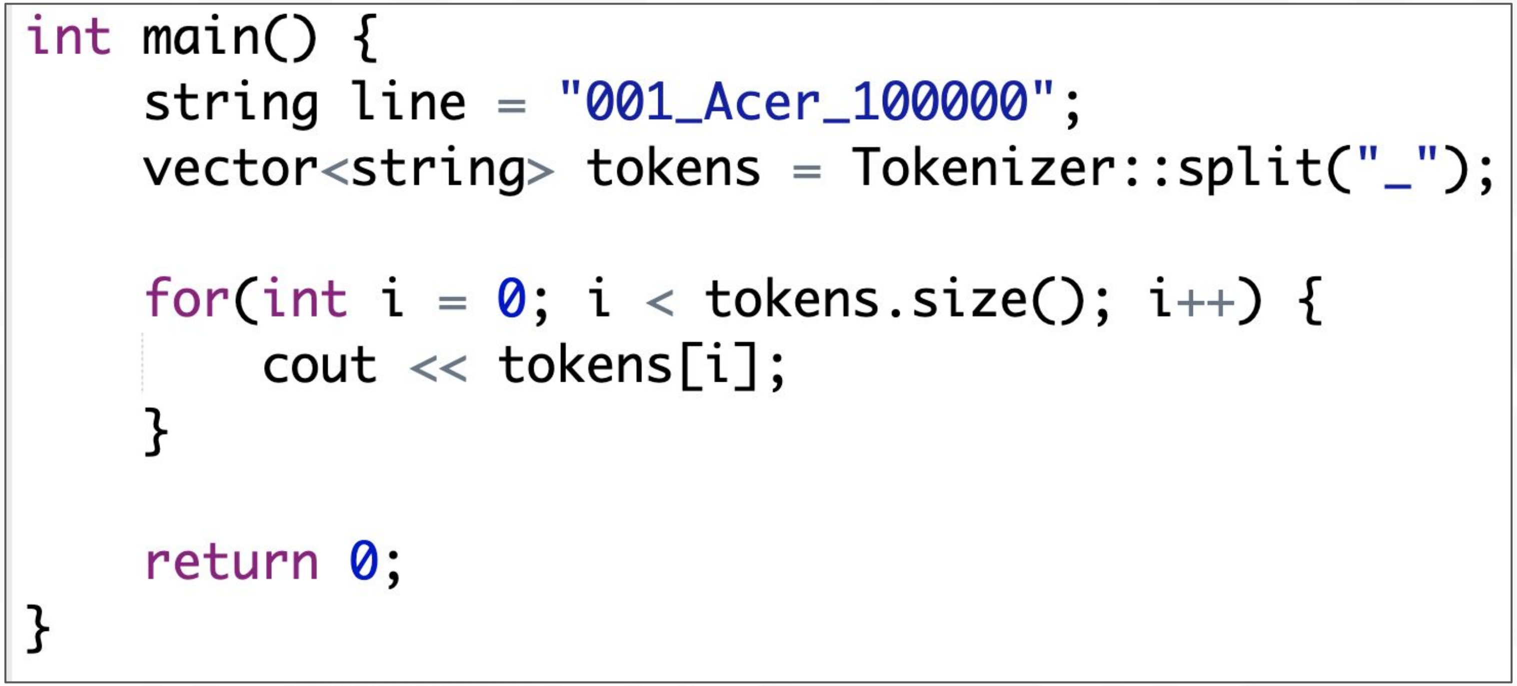
String split

Example: Finding substring

```
int main() {
    string haystack = "hello world war 3";
    string needle = "w";

    int startPosition = 0; // Vi tri bat dau tim, 0 la dau chuoi
    size_t foundPosition = haystack.find(needle, startPosition);

    if (foundPosition != string::npos) {
        cout << foundPosition << endl; // 6
    }

    startPosition = 7;
    foundPosition = haystack.find(needle, startPosition);

    if (foundPosition != string::npos) {
        cout << foundPosition << endl; // 12
    }

    cout << haystack.substr(13, 15); // ar 3
}
```

05. Project Tokenizer

Write a class that help split a string into vector of string

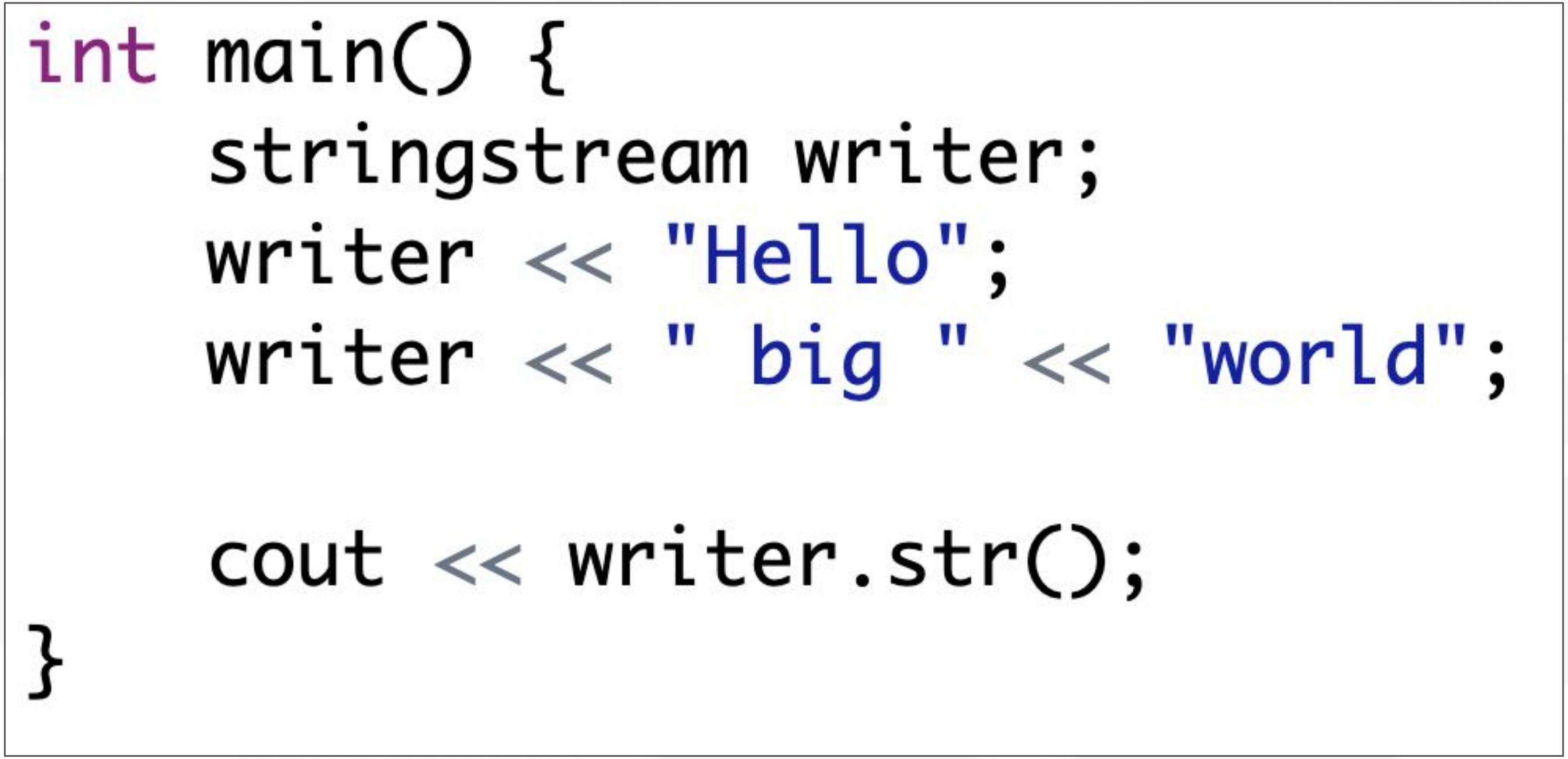


```
int main() {
    string line = "001_Acer_100000";
    vector<string> tokens = Tokenizer::split("_");

    for(int i = 0; i < tokens.size(); i++) {
        cout << tokens[i];
    }

    return 0;
}
```

```
001
Acer
10000
```

Fake Name

Example: Combining string



```
int main() {
    stringstream writer;
    writer << "Hello";
    writer << " big " << "world";

    cout << writer.str();
}
```

06. Project FakeName

- ❑ 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*

FakeName is the **business** class, for **generating data** using next

Fullname FakeName::next()

Fake Address

07. Project FakeHcmAddress

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

In main function, generate 20 fake HCM addresses

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.

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

Address FakeHcmAddress::next()

Fake Tel

08. Project FakeVnTel

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

In main function, generate 20 fake VN tels

Note: Display of tel is different like 0909 222 888

Hint

Tel is just a string, no need for a class

Fake Email

09. Project FakeEmail

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

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

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

In main function, generate 20 fake emails.

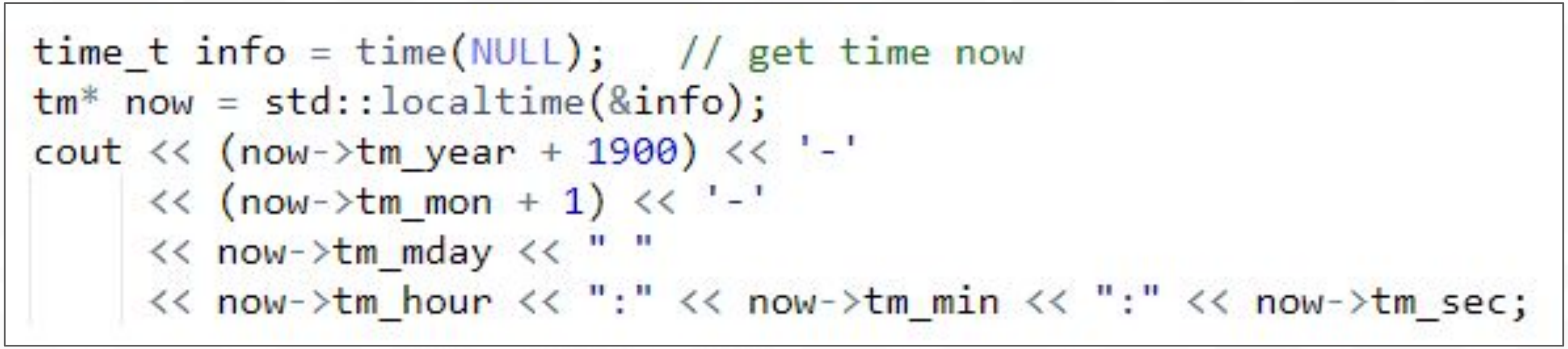
Hint

Email is just a string, no need for a class

Time

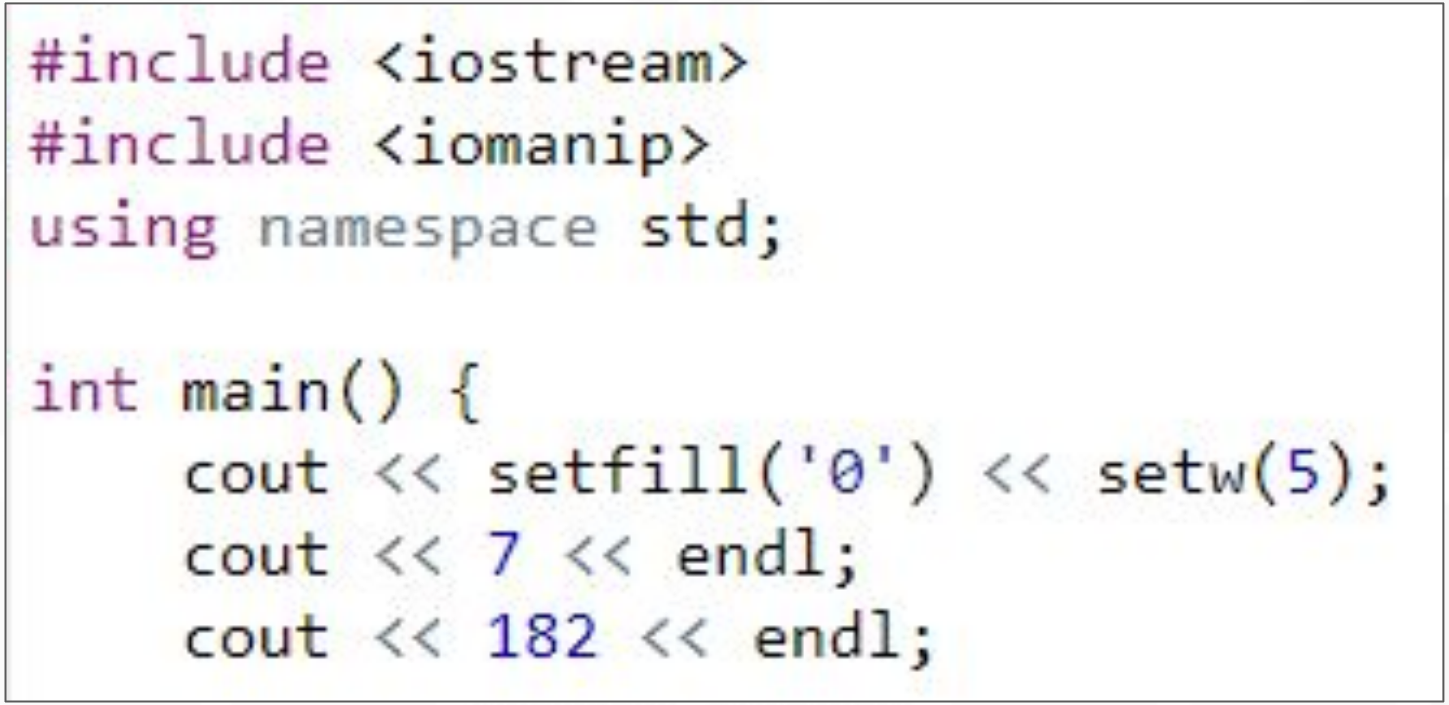
Example Get current time

#include <ctime>



```cpp
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
```

10. 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)

Requirement

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

What adjustment would you make to output time in 24 hour format (23:59:14) or in 12 hour format (11:59:14 PM)?

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

12. FakeBirthday

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 FakeBirthday::next()
```

FakeBirthday enhancement

Date **FakeBirthday**::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)

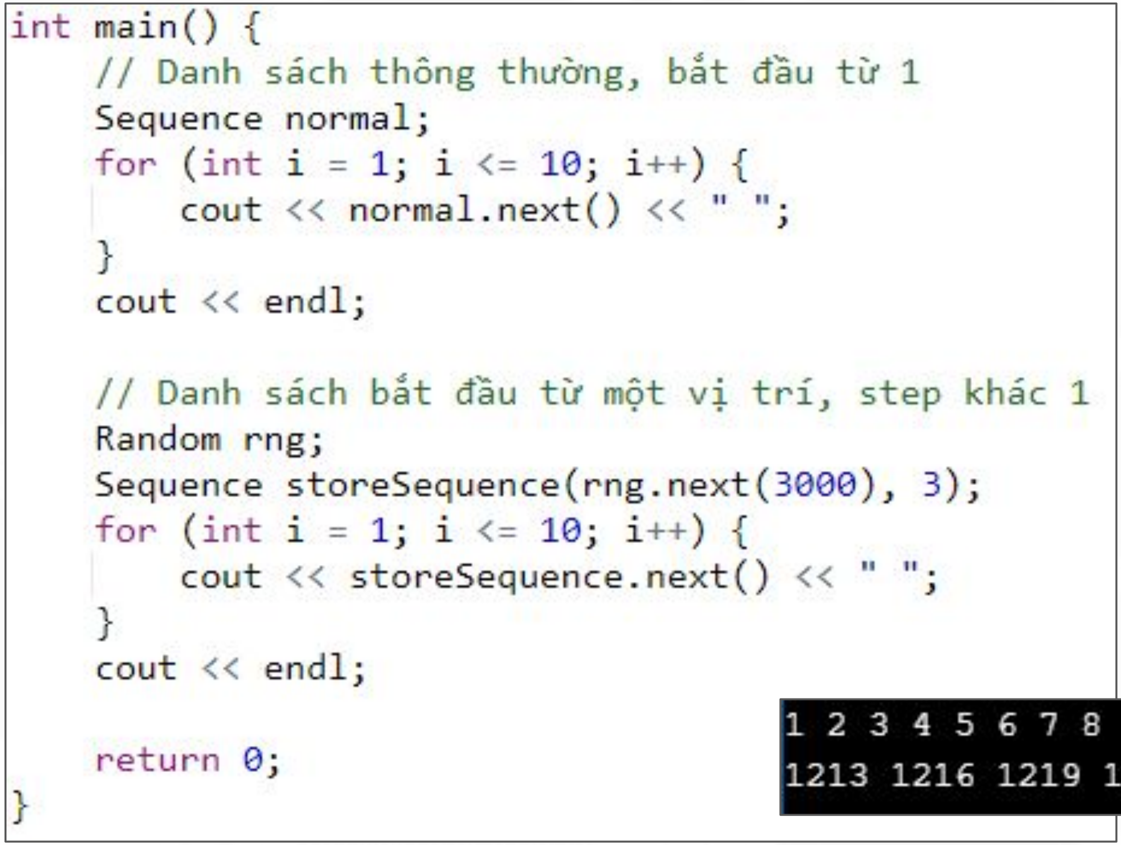
Sequence

13. Project Sequence

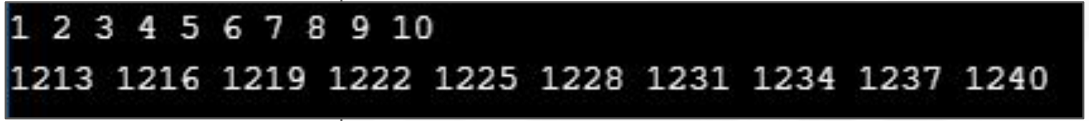
This is a common problem from

- ❑ Relational database like MySQL, Postgres for creating id
- ❑ Hiding actual bills number from a store within a day
  - ➢ Suppose the first bill is numbered 1, the next is 2
  - ➢ If we look at a bill and see its number is 398, we can assume easily that this store has roughly 398 transactions, some store owners want to hide this
  - ➢ The bills must be in ascending order for the **kitchen** to know everything is okay, no bill is dropped

Requirement



```cpp
int main() {
    // Danh sách thông thường, bắt đầu từ 1
    Sequence normal;
    for (int i = 1; i <= 10; i++) {
        cout << normal.next() << " ";
    }
    cout << endl;

    // Danh sách bắt đầu từ một vị trí, step khác 1
    Random rng;
    Sequence storeSequence(rng.next(3000), 3);
    for (int i = 1; i <= 10; i++) {
        cout << storeSequence.next() << " ";
    }
    cout << endl;

    return 0;
}
```



```
1 2 3 4 5 6 7 8 9 10
1213 1216 1219 1222 1225 1228 1231 1234 1237 1240
```