



# **LAB 3**

## **Object Oriented Design**

*Xuejing Lei*



# Outline

- Problem Description
- OOP Basics
- Data Structure
- Linear Search
- Read from and Write on files



# Problem Description

- **N Numbers:** 1, 2, 3, ... N
- **Subset:** K ( $0 < K \leq N$ ) numbers picked from N numbers.
- **Problem:** given M subsets, check whether any two subsets are independent
- **Fixed M=20, N=10**



# Problem Description

- **Two subsets are independent:**
  - all numbers in Set 1 are not in Set 2e.g.  $\{1, 2, 3\}$  and  $\{4, 8\}$  ✓  
 $\{1, 2, 3\}$  and  $\{1, 4\}$  ✗
- **Any two Subsets are independent:**
  - check all pairs of two sets



# Three Steps

1. Build class Number

OOP basics

2. Build class NumberSet

Data Structure,  
Linear Search

3. Write main Function

Read / Write Files  
Linear Search



# class Number

class Number



object: num\_1

Attributes

value

→ private attributes by default

Methods

get\_value

→ get values of private Attributes

set\_value

→ set / initialize values

# class NumberSet



```
// This is one possible way to define NumberSet class.  
// Other ways are also accepted.
```

```
class NumberSet {  
public:  
    // define as public attribute  
    Number num_array[10];  
  
    // no need to use constructor or destructor any more with public attributes  
    // public methods  
    bool check_independence(NumberSet obj) {  
        ... }  
}
```

## Note:

- If you use pointers, you may need Constructor/Destructor to allocate or delete memory.
- If you define attributes as private attributes, you may use Constructor & Destructor or `set_values(...)` & `get_values(...)` methods to do initialization.

# Data Structure



- **Array:**
  - must specify the size of a array when it is defined

In NumberSet class:

```
Number num_array[10]
```

In main function:

```
NumberSet set_array[20]
```



# Store Subset: Pointers vs. Array



- **Array:**

- must specify the size of a array when it is defined

Number array[size]

size = the largest  
possible size

- **Pointers:**

- can allocate memory somewhere else

Number \*array  
array = new Number[size]

size = the size of  
current NumberSet

# Linear Search



- **Search in an array**

Find if a given number is in an array

→ Iterate through the array from the first element, and compare each with the given number

- **NumberSet::check\_independence():**

Find whether two array are independent

→ Iterate over one array and search in the other array (two loops)

- **In main function:**

Find whether **any** two array are independent

→ Iterate over M arrays and call check\_independence in the inner loop (two loops)

# Read from File



- **ifstream**
  - Stream class to read from file
  - Example:

Read “1 2 3” from file and assign them to x, y, z

```
#include <fstream>
int x, y, z;
ifstream infile;
infile.open("input.txt");
infile >> x >> y >> z;
infile.close();
```

Or use this function to read a line:

```
getline (istream& is, string& str, char delim);
```

# Write on File



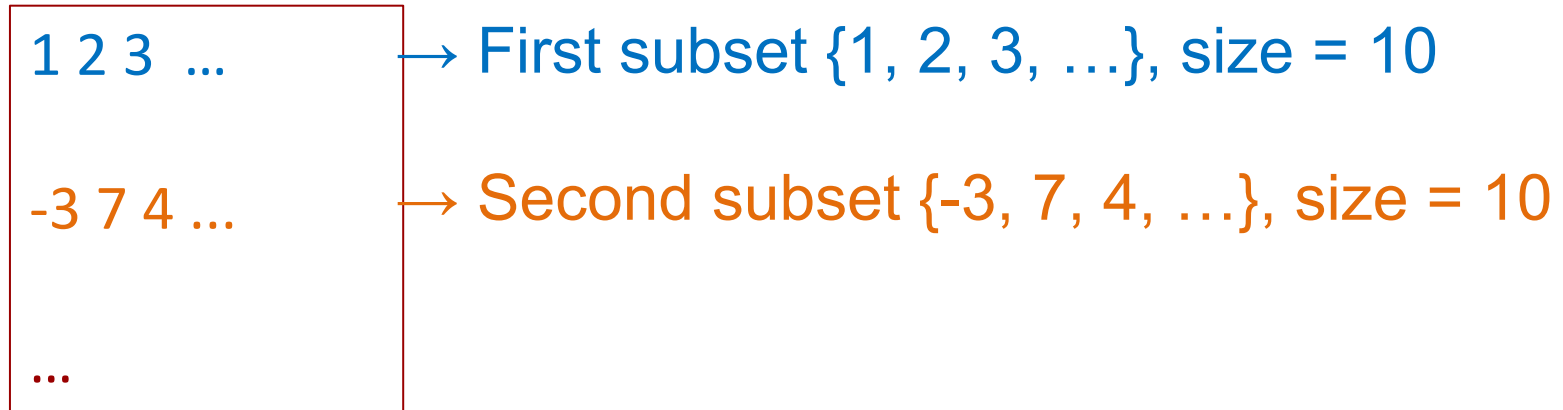
- **ofstream**
  - Stream class to write on file
  - Example:

Write x, y, z on file as

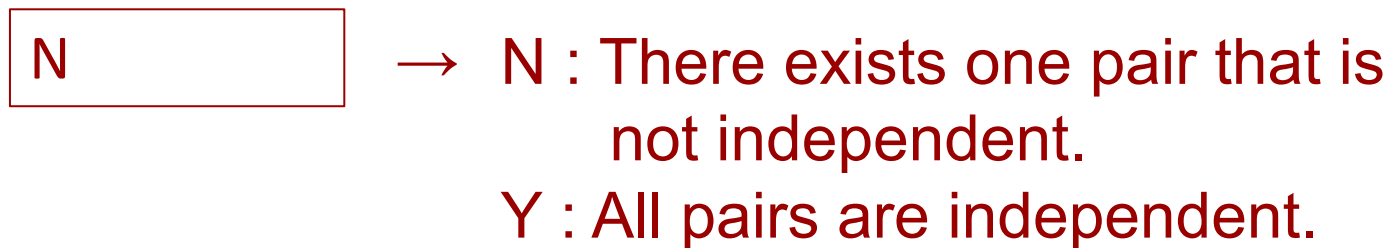
1	2
3	

```
#include <fstream>
int x=1, y=2, z=3;
ofstream outfile;
outfile.open("output.txt");
outfile << x << y << "\n";
Outfile << z;
outfile.close();
```

# Input File



# Output File





# Questions?