Practice 6
Class

2017 second semester
Computer Engineering Programming
Class

• Similar to structures
  – Structures : only member data
  – Classes : member data + member function

• Class Definition Example:

```cpp
class Student
{
public:
    string student_id;
    string name;
    int age;
    int score;
    void print_studentInfo();
}
```
Class Member Function Definition

• Member functions must be defined like other functions.
  – Can be after/before main() definition

• Must **specify class** for member function definition using “::”.

```cpp
void Student::print_studentInfo()
{
  ...
}
```
```cpp
#include <iostream>
#include <string>

using namespace std;

class Student {
public:
    string student_id;
    string name;
    int age;
    int score;
    void print_studentInfo();
};

int main()
{
    Student a;
    a.student_id = "2017312724";
    a.name = "KIM MIN SU";
    a.age = 20;
    a.score = 80;
    a.print_studentInfo();
    return 0;
}

void Student::print_studentInfo()
{
    cout<<student_id<<"\t"<<name<<"\t"<<age<<"\t"<<score<<"\n";
}
```
Public and Private Member

• **Public vs. Private**
  – Public items are “user-accessible”. Usually member function.
  – Private items only can be accessed by themselves class.

• **Member Data in class almost always designated private in definition for:**
  – Uphold principles of OOP
    • Information Hiding
    • Data Abstraction
    • Encapsulation
  – Hide data from user
  – Allow manipulation only via member functions.
Q. What happens if we access student_id variable in main function?

CHECK!
How to use string?

- **C++ use string library to save character stream.**
  - In C, we use char array to save string.
  - Instead of array, C++ provide string library to save and manipulated strings.
  - To use string library, you should include `<string>`

- More detail will learn later.
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s1 = "hello";
    string s2 = "world";
    string s3 = "hello";

    cout << "s1 : " << s1 << endl;
    cout << "s2 : " << s2 << endl;
    cout << "s3 : " << s3 << endl;

    if (s1.compare(s2) == 0)
        cout << "s1 and s2 is identical" << endl;
    else
        cout << "s1 and s2 is different" << endl;

    if (s1.compare(s3) == 0)
        cout << "s1 and s3 is identical" << endl;
    else
        cout << "s1 and s3 is different" << endl;

    return 0;
}
Exercise 1

- Write the program save 3 students’ name in C++ string array.
Exercise 1

- **Make & Use class StudentDB including :**
  - **Member Data**
    - string db[3] (string array type) // array to save student name.
    - Int total_student (int type) // amount of name in db array.

  - **Member Function**
    - int add_student(string name) // add new student.  // find student’s index when the parameter name is given.
    - int get_index_by_name(string name)
      // find student’s name when the parameter index is given.
    - string get_name_by_index(int index)
      // change db’s information on corresponding index.
    - int change_name_at_index(int index, string name)
#include <iostream>
#include <string>

using namespace std;

class StudentDB
{
public:
    int add_student(string name);
    int get_index_by_name(string name);
    string get_name_by_index(int index);
    int change_name_at_index(int index, string name);

private:
    string db[3];
    int total_student = 0;
};

int main()
{
    StudentDB std_db;
    std_db.add_student("dodo");
    std_db.add_student("momo");
    std_db.add_student("coco");
    std_db.add_student("popo");
    cout << std_db.get_index_by_name("mimi") << endl;
    cout << std_db.get_index_by_name("dodo") << endl;
    cout << endl;
    cout << std_db.get_name_by_index(4) << endl;
    cout << std_db.get_name_by_index(2) << endl;
    cout << endl;
    std_db.change_name_at_index(5, "mama");
    std_db.change_name_at_index(1, "yoyo");
    cout << std_db.get_name_by_index(1) << endl;
    cout << endl;
    return 0;
}"
Exercise 1

DB is FULL
-1
0

NO student at this index
coco

NO student at this index
yoyo