

## **EEEC10008(515169) S23: Object-Oriented Programming**

### **Pointer & Reference**



#### **What you will learn from Lab 2**

In this laboratory, you will review the usage of pointer and reference, which are important in object-oriented programming.

#### **TASK 2-1: POINTER TYPE**

- ✓ Program lab2-1 below shows some using examples for pointer manipulation including pointer declarations and assignments.

```
// lab2-1.cpp
#include <iostream>
using namespace std;

int main()
{
    double a = 1.34;
    double *pa = &a;

    cout << "  a = " <<  a << endl;
    cout << " &a = " <<  &a << endl;
    cout << " *a = " <<  *a << endl;
    cout << " pa = " <<  pa << endl;
    cout << "&pa = " << &pa << endl;
    cout << "*pa = " << *pa << endl;

    *pa = 6.5;
    cout << "a = " << a << endl;
    cout << "*pa = " << *pa << endl;

    return 0;
}
```

- ☐ Please try to explain the execution results by yourself.
- ☐ Note that there is a *compiler error* in this example.

#### **TASK 2-2 : REFERENCE TYPE**

- ✓ A reference is an implicit pointer that is automatically dereferenced.

```
// lab2-2.cpp
#include <iostream>
using namespace std;

int main()
{
    int a = 1024;
    int &refa = a;

    cout << "    a = " <<  a << endl;
    cout << "  &a = " <<  &a << endl;
    cout << "   *a = " <<  *a << endl;
    cout << " refa = " << refa << endl;
}
```

```
    cout << "&refa = " << &refa << endl;
    cout << "*refa = " << *refa << endl;

    return 0;
}
```

- ☐ Please try to explain the execution results by yourself.
- ☐ Note that there is a *compiler error* in this example.
- ☐ The addresses of `pa` and `a` are different in program lab2-1, but the addresses of `refa` and `a` are the same in this example.

### TASK2-3: POINTER TO CONSTANT

- ✓ Please differentiate the following three examples (*a-point-to-a-constant*, *a-constant-pointer* and *a-constant-pointer-to-a-constant*) and fix the compiler errors with improper usage.

```
// lab2-3-1.cpp
#include <iostream>
using namespace std;

int main()
{
    double a = 1.34;
    const double *pa = &a; // a pointer to a constant

    cout << "*pa = " << *pa << endl;

    double b = 6.5;
    pa = &b;                // a pointer to a constant can change the pointer
    cout << "*pa = " << *pa << endl;

    *pa = 7.6;              // cannot modify a pointer to a constant
    cout << "*pa = " << *pa << endl;

    return 0;
}
```

```
// lab2-3-2.cpp
#include <iostream>
using namespace std;

int main()
{
    double a = 1.34;
    double *const pa = &a; // a const pointer to a double

    cout << "*pa = " << *pa << endl;

    double b = 6.5;
    pa = &b;                // a constant pointer cannot be changed
    cout << "*pa = " << *pa << endl;

    *pa = 7.6;              // a constant pointer can be modified
    cout << "*pa = " << *pa << endl;
}
```

```
    return 0;
}
```

```
// lab2-3-3.cpp
#include <iostream>
using namespace std;

int main()
{
    double a = 1.34;
    const double *const pa = &a; // a const pointer to a constant

    cout << "*pa = " << *pa << endl;

    double b = 6.5;
    pa = &b; // a constant pointer to constant cannot be
    changed
    cout << "*pa = " << *pa << endl;

    *pa = 7.6; // a constant pointer to constant cannot be
    modified
    cout << "*pa = " << *pa << endl;

    return 0;
}
```

## TASK2-4: STRUCTURE INITIALIZATION

- ✓ Program lab2-4.cpp shows how to initialize struct.

```
// lab2-4.cpp
#include <iostream>

using namespace std;

struct student
{
    string name;
    int mathScore;
    int chineseScore;
};

int main(int argc, char *argv[])
{
    struct student students[3] = {
        {.name = "Bob", .mathScore = 100, .chineseScore = 90},
        {.name = "Alice", .mathScore = 59, .chineseScore = 59},
        {.name = "John", .mathScore = 87, .chineseScore = 87},
    };

    for(int i = 0; i < 3; i++)
    {
```

```
        cout << "studentName: " << students[i].name << endl;
        cout << "\t math score: " << students[i].mathScore << endl;
        cout << "\t chinese score: " << students[i].chineseScore << endl;
    }

    return 0;
}
```

## EXERCISE 2-1

- ✓ In this exercise, you have to implement a vending machine with the `class` “VM”. Besides, you also have to use the `struct` “item” to store item information.
- ✓ The following are the variables in the `struct` “item”.
  - id**(int)
  - name**(string)
  - price**(int)
- ✓ The following are the variables and functions in the `class` VM.
  - Private Variables:
    - **item\_vec**(vector<item>)
    - **password**(string)
  - Private Functions:
    - void **add\_item**(): being called in `print_optionList()`
    - void **remove\_item**(): being called in `print_optionList()`
    - void **find\_change**(item, int, int, int, int): being called in `purchase_item()` and the four int parameters are the amount of one, five, ten and fifty dollar coins
    - void **purchase\_item**(): being called in `print_optionList()`
    - void **print\_itemList**(): being called in `print_optionList()` and the list have to be sorted by “id” ascendingly
    - void **print\_optionList**(): repeatedly being called in `start()`
    - string **set\_password**(): being called in `start()` and return the password
  - Public Functions:
    - void **start**(): being called in the main function to start the vending machine
- ✓ Please write the definition of `struct` “item” and the declaration of `class` “VM” in the file “VM.h”, write the definition of `class` “VM” functions in the file “VM.cpp”, and run the program in the file “lab2.cpp”.
- ✓ The amount of the change (including one, five, ten and fifty dollars) in the vending machine is assumed to be infinite.
- ✓ Do not have to consider the case that the payment is not enough, that is, customers always insert money that is equal or higher than the item price.

- ✓ Each item has a unique ID and there are no two items with the same name (quantity of every item is zero or one).
- ✓ Execution Result

```
$ ./lab2
Enter a password for your vending machine: 123
Enter your password again: 12345
It's different from the first password! Please set the password again!
Enter a password for your vending machine: 1234
Enter your password again: 1234
Enter 1 for owner mode, 2 for customer mode: 3
Invalid input!
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
1
Please enter the information of the item.
ID: 6
Name: apple
Price: 17
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
1
Please enter the information of the item.
ID: 3
Name: grape
Price: 31
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
1
Please enter the information of the item.
ID: 5
Name: orange
Price: 63
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
```

```
1. Add item
2. Remove item
3. Print item list
3
ID      Name    Price
3       grape   31
5       orange  63
6       apple   17
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
2
The name of the item you want to remove: banana
No item found!
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
2
The name of the item you want to remove: orange
Enter 1 for owner mode, 2 for customer mode: 2
Welcome! Here is the item list.
ID      Name    Price
3       grape   31
6       apple   17
Which item do you want to buy?
orange
Out of stock!
Enter 1 for owner mode, 2 for customer mode: 2
Welcome! Here is the item list.
ID      Name    Price
3       grape   31
6       apple   17
Which item do you want to buy?
apple
The price is 17 dollars
Please insert coins.
One dollar: 2
Five dollar: 1
Ten dollar: 1
Fifty dollar: 0
No change should be given.
Thank you for your purchase!
```

```
Enter 1 for owner mode, 2 for customer mode: 2
Welcome! Here is the item list.
ID      Name      Price
3       grape     31
Which item do you want to buy?
grape
The price is 31 dollars
Please insert coins.
One dollar: 4
Five dollar: 1
Ten dollar: 4
Fifty dollar: 1
Your change
One dollar: 3
Five dollar: 1
Ten dollar: 1
Fifty dollar: 1
Thank you for your purchase!
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 1234
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
3
There is no item right now!!!
Enter 1 for owner mode, 2 for customer mode: ^C
```

- ✓ Typing the wrong password in owner mode should be handled as follow

```
$ ./lab2
Enter a password for your vending machine: 123
Enter your password again: 123
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 12345
Wrong password!
Enter 1 for type again, 2 for going back to last step: 2
Enter 1 for owner mode, 2 for customer mode: 1
Please enter your password: 12345
Wrong password!
Enter 1 for type again, 2 for going back to last step: 1
Please enter your password: 123
Welcome to owner mode!
1. Add item
2. Remove item
3. Print item list
^C
```

- ✓ Entering the customer mode when there is no item yet

```
$ ./lab2
Enter a password for your vending machine: 123
Enter your password again: 123
Enter 1 for owner mode, 2 for customer mode: 2
Sorry! There is no item right now.
Enter 1 for owner mode, 2 for customer mode: ^C
```