

UEE1302(1066) F12: Introduction to Computers and Programming

Lab 8: User-Defined Data Type & Array & C-String



What you will learn from Lab 8

In this laboratory, you will understand how to customize user-defined data type with enum, struct.

TASK 8-1 : ENUM

- ✓ Please predict the result of program lab8-1 and then execute this program to compare the result on screen with what you predict.

```
//File: lab8-1.cpp
#include <iostream>
using namespace std;

enum baseball {FIRSTBAT = 1, SECONDBAT = 2, THIRDBAT = 3, FOURTHBAT = 4};

int main()
{
    baseball BatOrder = baseball(2);
    if (BatOrder == THIRDBAT)
    {
        cout << "The BatOrder is THIRDBAT" << endl;
    }

    return 0;
}
```

- Try to modify the program if there is no output on screen.

TASK 8-2 : STRUCTURE

- ✓ Example *struct*: please predict the result of program lab8-2 and then execute this program to compare the result on screen with what you predict.

```
//File: lab8-2.cpp
#include <iostream>
using namespace std;

struct Time
{
    int hours;
    int minutes;
    int seconds;
};

void ShowSeconds(const Time &now);
Time SetTime();

int main()
{
```

```
    Time t = {0, 0, 0};  
    t = SetTime();  
  
    ShowSeconds(t);  
  
    return 0;  
}  
  
Time SetTime()  
{  
    Time now;  
    cout << "Please Enter Current Time:" << endl;  
    cout << "Current hour:";  
    cin >> now.hours;  
    cout << "Current minute:";  
    cin >> now.minutes;  
    cout << "Current second:";  
    cin >> now.seconds;  
  
    return now;  
}  
  
void ShowSeconds(const Time &now)  
{  
    int seconds = 0;  
    seconds = now.hours * 3600 + now.minutes * 60 + now.seconds;  
    cout << "Total seconds of today: " << seconds << endl;  
  
    now.hours = 12;  
    seconds = now.hours * 3600 + now.minutes * 60 + now.seconds;  
    cout << "Total seconds of new: " << seconds << endl;  
}
```

- Type 1 structure: use struct only
struct struct_tag
{
 struct body;
};
- The keyword **const** can be added to the declaration of an object to make the values of the object cannot be changed.
For example: void ShowSeconds(const Time &now) means that the value of the Time object cannot be modified in the ShowSeconds function.
- A struct object can initialize at declaration
For example: Time t = {0, 0, 0};
- Use dot operator (.) to access members.
For example: seconds = now.hours * 3600 + now.minutes * 60 + now.seconds;

TASK 8-3 : ARRAY AS ARGUMENT

- ✓ *Sorting* is used to sort an array with increasing order or decreasing order. Please predict the result of program lab8-3

```
//File: lab8-3.cpp
#include <iostream>
using namespace std;

void DisplayArray(int [],int);
void InsertionSort(int [],int);

int main()
{
    const int MaxSize = 10;
    int array[MaxSize] = {21,15,12,24,9,30,27,6,3,18};
    DisplayArray(array,MaxSize);
    InsertionSort(array,MaxSize);
    DisplayArray(array,MaxSize);
}

void DisplayArray(int vec[], int size)
{
    for (int idx = 0; idx < size;idx++)
    {
        cout << vec[idx] << " ";
    }
    cout << endl;
}

void InsertionSort(int vec[],int size)
{
    for (int jdx = 1; jdx < size; jdx++)
    {
        int key = vec[jdx];
        int idx = jdx - 1;

        while ( idx >= 0 && vec[idx] > key)
        {
            vec[idx+1] = vec[idx];
            --idx;
        }
        vec[idx+1] = key;
    }
}
```

TASK 8-4 : TWO-DIMENSIONAL ARRAY

- ✓ Please predict the result of program lab8-4, and then execute this program to compare the result on screen with what you predict.

```
//File: lab8-4.cpp
#include <iostream>
#include <iomanip>
using namespace std;

const int numrows = 3;
const int numcols = 4;

int main()
{
    int val[numrows][numcols]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    cout << "Display the multiplied elements";
    for (int idx = 0; idx < numrows; idx++)
    {
        cout << endl;
        for ( jdx = 0; jdx < numcols; jdx++)
        {
            val[idx][jdx] = val[idx][jdx] * 10;
            cout << setw(5) << val[idx][jdx];
        }
        cout << endl;
    }
    return 0;
}
```

- ✓ Please predict the result of program lab8-5, and then execute this program to compare the result on screen with what you predict.

```
//File: lab8-5.cpp
#include <iostream>
#include <iomanip>
using namespace std;

const int numrows = 3;
const int numcols = 4;
void fun(int [][]);
int main()
{
    int val[numrows][numcols]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    fun(val);
    return 0;
}
void fun(int vec[][])
```

```
{
    cout << "Display the multiplied elements";
    for (int idx = 0; idx < numrows; idx++)
    {
        cout << endl;
        for ( jdx = 0; jdx < numcols; jdx++)
        {
            vec[idx][jdx] = vec[idx][jdx] * 10;
            cout << setw(5) << val[idx][jdx];
        }
        cout << endl;
    }
}
```

- Modify the problem if any compiler error exists.

TASK 8-5 : C-STRING

- ✓ Please predict the result of program lab8-6, and then execute this program to compare the result on screen with what you predict.

```
//File: lab8-6.cpp
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char stringA[20] = {'C','+', '+', '\0'};
    char stringB[20] = "Programming";

    cout << "length of " << stringA << " is " << strlen(stringA) << endl;
    cout << "length of " << stringB << " is " << strlen(stringB) << endl;

    char stringC[20];

    // Copy stringA to stringC
    strcpy(stringC, stringA);
    cout << stringC << endl;

    // Concatenate " " and stringB to stringC
    strcat(stringC, " ");
    strcat(stringC, stringB);
    cout << stringC << endl;

    return 0;
}
```

- ✓ Please predict the result of program lab8-7, and then execute this program to compare the result on screen with what you predict.

```
//File: lab8-7.cpp
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char str1[80];
    char str2[80];

    cout << "Enter a sentence: ";
    cin.getline(str1,80);

    cout << "Enter another sentence: ";
    cin.getline(str2,80);

    if (strcmp(str1,str2))
    {
        cout << str1 << " != " << str2 << endl;
    }
    else
    {
        cout << str1 << " == " << str2 << endl;
    }

    return 0;
}
```

TASK 8-6 : EXERCISES

- ✓ type includes student id, gender, age, and the amount of cash a student owns. The struct type should be applied to record for such members, i.e.

```
struct student{  
    long id;  
    char gender;  
    int age;  
    int money;  
};
```

Assume that there are five students in this student group. Information of four students can be initialized as follows,

```
student a = {9413599, 'M', 25, 10001};  
student b = {9513622, 'M', 22, 500};  
student c = {9613812, 'F', 20, 30};  
student d = {9713964, 'F', 18, 66000};
```

Please record your own information via input from the keyboard (cin) and then display the information of each group member according to his/her record. “Youth” denotes that the age of one student is smaller than 21 while “Adult” denotes to the student is larger than or equal to 21 years old. “Rich” refers to the case that the student has more than \$10000 while “poor” refers to the remaining. “Male/Female” is the gender of the student. Describe each member by a complete sentence instead of the record data only.

The required format is shown as follows.

```
> ./ex8-1  
Please enter your student id: 9823016  
Gender (M/F): M  
Age: 18  
The amount of money in your own: 30000  
id 9413599 is a adult & rich Male!  
id 9513622 is a adult & poor Male!  
id 9613812 is a young & poor Female!  
id 9713964 is a young & rich Female!  
id 9823016 is a young & rich Male!  
This group is 20.6 years old in average.  
This group has $106531 now.  
>
```

- ✓ Please write a program which can generate a matrix randomly with specific number of rows and columns. Note that the number of rows and columns are less than 10 and the range of the

elements in matrix is (0, 20]. The modified matrix is composed by sorting the elements for each row. You can use the `InsertionSort()` in problem lab9-1. However, the function is used to sort an array with increasing order, but you should modify `InsertionSort()` to sort with decreasing order. The required format is shown as follows.

```
> ./ex8-2
Please enter the number of rows: 3
Please enter the number of columns: 5
The original matrix is
10 8 4 5 11
3 9 20 7 18
5 6 17 4 9
The modified matrix is
11 10 8 5 4
20 18 9 7 3
17 9 6 5 4
>
```

- The program should follow the following architecture.

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

const int MaxSize = 10;
void DisplayMatrix(const int [][][MaxSize],const int,const int);
void GenerateMatrix(int [][][MaxSize],const int,const int);
void SortMatrix(int [][][MaxSize],const int,const int);
void InsertionSort(int [],const int);

int main()
{
    int numrows,numcols;
    int IntMatrix[MaxSize][MaxSize];

    cout << "Please enter the number of rows: ";
    cin >> numrows;
    cout << "Please enter the number of columns: ";
    cin >> numcols;

    cout << "The original matrix is" << endl;
    GenerateMatrix(IntMatrix,numrows,numcols);
    DisplayMatrix(IntMatrix,numrows,numcols);
    SortMatrix(IntMatrix,numrows,numcols);
    cout << "The modified matrix is" << endl;
    DisplayMatrix(IntMatrix,numrows,numcols);
}
```

Exercise after LAB

1. Present Value

Suppose you want to deposit a certain amount of money in to a savings account, and then leave it alone to draw interest for the next 10 years. At the end of 10 years you would like to have \$10,000 in the account. How much do you need to deposit today to make that happen? To find out you can use the following formula, which is known as the present value formula:

The terms in the formula are as follows:

- P is the present value, or the amount that you need to deposit today.
- F is the future value that you want in the account (in this case, \$10,000).
- r is the annual interest rate (expressed in decimal form, such as 0.042).
- n is the number of years that you plan to let the money sit in the account.

Write a program with a function named `presentValue` that performs this calculation. The function should accept the future value, annual interest rate, and number of years as arguments. It should return the present value, which is the amount that you need to deposit today. Demonstrate the function in a program that lets the user experiment with different values for the formula's terms.

2. Falling Distance

The following formula can be used to determine the distance an object falls due to gravity in a specific time period:

The variables in the formula are as follows: d is the distance in meters, g is 9.8, and t is the time in seconds that the object has been falling.

Write a function named `fallingDistance` that accepts an object's falling time (in seconds) as an argument. The function should return the distance, in meters, that object has fallen during that time interval. Write a program that demonstrates the function by calling it in a loop that passes the values 1 through 10 as arguments, and display the return value