

UEE1302(1066) F12: Introduction to Computers and Programming

Function (II) - Parameter



What you will learn from Lab 7

In this laboratory, you will understand how to use typical function prototype with call-by-value concept and the concept of local and global variables.

TASK 7-1 : PARAMETER PASSING

- ✓ Call-by-value example: Please predict the result of program lab7-1, and then execute this program to compare the result on screen with what you predict.

```
//File: lab7-1.cpp
#include <iostream>
using namespace std;

void swap(int, int);

int main()
{
    int x = 5, y = 6;
    cout << "x = " << x << " and y = " << y << endl;
    swap(x, y);
    cout << "x = " << x << " and y = " << y << endl;
    return 0;
}

void swap(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

- It is noted that the swap function is utilized to exchange the two values with each other. However, the values of x and y in the main function do not swap. What observation can be concluded from the usage of call-by-value?
- Below program lab7-2 and program lab7-3 are two kinds of solutions to solve the problem.

- ✓ Globe Variable example: Please predict the result of program lab7-2, and then execute this program to compare the result on screen with what you predict.

```
//File: lab7-2.cpp
#include <iostream>
using namespace std;
┌
void swap();
int x = 5, y = 6;
int main()
{
    cout << "x = " << x << " and y = " << y << endl;
    swap();
    cout << "x = " << x << " and y = " << y << endl;
    return 0;
}
void swap()
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

- It is noted that no parameter passing is performed in the program. The swap function is simply utilized to exchange the two values with each other. Since the variables of x and y are global variables in the program, the values are swapped. What observation can be concluded from the usage of global variables?

- ✓ Call-by-Reference example: Please predict the result of program lab7-3, and then execute this program to compare the result on screen with what you predict.

```
//File: lab7-3.cpp
#include <iostream>
using namespace std;
┌
void swap(int&, int&);
int main()
{
    int x = 5, y = 6;
    cout << "x = " << x << " and y = " << y << endl;
```

```
    swap(x, y);  
    cout << "x = " << x << " and y = " << y << endl;  
    return 0;  
}  
void swap(int& x, int & y)  
{  
    int temp;  
    temp = x;  
    x = y;  
    y = temp;  
}
```

- What observation can be concluded from the usage of call-by-reference?

TASK 7-2 : FUNCTION INPUT & OUTPUT

- ✓ Return-Type output: Please predict the result of program lab7-4, and then execute this program to compare the result on screen with what you predict.

```
//File: lab7-4.cpp  
#include <iostream>  
using namespace std;  
└─  
int fun(int, int);  
int main()  
{  
    int a = 1, b = 1, c = 1;  
    c = fun(a, b);  
    cout << "The value returned is " << c << endl;  
    return 0;  
}  
int fun(int x, int y)  
{  
    return (x+y);  
}
```

- ✓ Call-by-Reference Type output: Please predict the result of program lab7-5, and then execute this program to compare the result on screen with what you predict.

```
//File: lab7-5.cpp
#include <iostream>
using namespace std;
↵
void sumavg(double, double, double, double, double);
int main()
{
    double n1, n2, n3, sum, avg;
    sum = 0.0;
    avg = 0.0;
    cout << "Please enter score of three students" << endl;
    cin >> n1;
    cin >> n2;
    cin >> n3;
    sumavg(n1, n2, n3, sum, avg);
    cout << "The total score of three students is " << sum << endl;
    cout << "The average score of three students is " << avg << endl;
    return 0;
}
void sumavg(double n1, double n2, double n3, double sum, double avg)
{
    sum = n1 + n2 + n3;
    avg = (n1 + n2 + n3) / 3;
}
```

- Modify the prototype of function sumavg as void sumavg(double, double, double , double &, double &)

TASK 7-3 : RECURSION

- ✓ Lab7-6 is an example code for recursive. Please write a C++ program to ask the user to enter an integer i and calculate the answer for the following equation:

$$f(i) = \begin{cases} i/2 & \text{if } i \text{ is even} \\ f(f(3i+1)) & \text{elsewhere} \end{cases}$$

```
//File: lab7-6.cpp
#include <iostream>

using namespace std;

int ff(int);

int main()
{
    cout << "Enter i: ";
    int i;
    cin >> i;
    cout << "f(" << i << ") = " << ff(i) << endl;
    return 0;
}

int ff(int x)
{
    if (x%2==0)
        return x/2;
    else
        return ff(ff(3*x+1));
}
```

- The required format is shown as follows.

(Ex) >./lab7-6

Enter i: 4 ↵

f(4)= 2

TASK 7-4 : EXERCISES

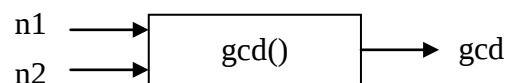
- ✓ Write a C++ program to convert the decimal number to the binary number. You should write a function to do the conversion. Please meet the format requirements as follows.

```
> ./ex7-1
Enter a decimal value:
17
The binary value of 17 is 10001
Enter a decimal value:
1024
The binary value of 1024 is 10000000000
Enter a decimal value:
-1
Exit!!
>
```

- Note: Please use data type “long” to declare the variable(s) of the input variable and output sequence.
 - Note: If input is less than 0, exit.
- ✓ Design a system which can calculate the greatest common divisor (g.c.d.) and least common multiple (l.c.m.) of 5 integers. Please meet the requirements as follows.

```
> ./ex 7-2
Input five integer number:
200 500 800 300 700
g.c.d. : 100
l.c.m. : 84000
>
```

- Your function prototype should follow the structure:



And please use recursive function to do this:

For $n1 \geq n2 > 0$, $\text{gcd}(n1, n2) = \begin{cases} n2 & \text{if } n2 \text{ divided with no remainder} \\ \text{gcd}(n2, n1 \% n2) & \text{otherwise} \end{cases}$

- Try to put the code of calculating the g.c.d. part and l.c.m. into a function. Then you can call the function for four times instead of writing the same codes again and again.