

5-1 Functions:

A function is a set of statements designed to accomplish a particular task. Experience has shown that the best way to develop and maintain a large program is to construct it from smaller pieces or (modules). Modules in C++ are called functions.

General Form of Function:

```
return-type function-name ( parameters-list )  
{  
    statement1 ;  
    statement2 ;  
    :  
    statement-n ;  
}
```

Example 1:

```
void printmessage ( )  
{  
    cout << "University of Technology";  
}  
  
void main ( )  
{  
    printmessage();  
}
```

Example 2:

```
int max (int a, int b)  
{  
    int c;  
    if (a > b) c = a;  
    else c = b;  
    return (c);  
}  
  
void main ( )  
{  
    cout << max (5, 6);  
}
```

Example 5.1

 Write C++ program to calculate the squared value of a number passed from main function. Use this function in a program to calculate the squares of numbers from 1 to 10:

```
#include<iostream.h>
int square ( int y )
{
    int z;
    z = y * y;
    return ( z );
}

void main( )
{
    int x;
    for ( x=1; x <= 10; x++ )
        cout << square ( x ) << endl;
}
```

Example 5.2

 Write C++ program using function to calculate the average of two numbers entered by the user in the main program:

```
#include<iostream.h>
float aver (int x1, int x2)
{
    float z;
    z = ( x1 + x2 ) / 2.0;
    return ( z );
}

void main( )
{
    float x;
    int num1,num2;
    cout << "Enter 2 positive number \n";
    cin >> num1 >> num2;
    x = aver (num1, num2);
    cout << x;
}
```

Example 5.3

 Write C++ program, using function, to find the summation of the following series:

$$\sum_{i=1}^n i^2 = 1^2 + 2^2 + 3^2 + \dots + n^2$$

```
#include<iostream.h>
int summation ( int x)
{
    int i = 1, sum = 0;
    while ( i <= x )
    {
        sum += i * i;
        i++;
    }
    return (sum);
}
void main ( )
{
    int n ,s;
    cout << "enter positive number";
    cin >> n;
    s = summation ( n );
    cout << "sum is: " << s << endl;
}
```

Example 5.4:

 Write a function to find the largest integer among three integers entered by the user in the main function.

```
#include <iostream.h>
int max(int y1, int y2, int y3)
{
    int big;
    big=y1;
    if (y2>big) big=y2;
    if (y3>big) big=y3;
    return (big);
}
void main( )
{
    int largest,x1,x2,x3;
    cout<<"Enter 3 integer numbers:";
    cin>>x1>>x2>>x3;
    largest=max(x1,x2,x3);
    cout<<largest;
}
```

Exercise:

- (1) Write C++ program, using function, to inverse an integer number:
For example: 765432 → 234567
- (2) Write C++ program, using function, to find the summation of student's marks, and it's average, assume the student have 8 marks.

5-2 Passing Parameters:

There are two main methods for passing parameters to a program:

(1) **passing by value**, and (2) **passing by reference**.

A- Passing by value:

When parameters are passed by value, a copy of the parameters value is taken from the calling function and passed to the called function. The original variables inside the calling function, regardless of changes made by the function to it are parameters will not change. All the previous examples used this method.

B- Passing by Reference:

When parameters are passed by reference their addresses are copied to the corresponding arguments in the called function, instead of copying their values. Thus pointers are usually used in function arguments list to receive passed references.

This method is more efficient and provides higher execution speed than the call by value method, but call by value is more direct and easy to use.

Example 5.5:



The following program illustrates passing parameter by reference.

```
#include <iostream.h>
void swap(int *a,int *b)
{
    int t;
    t=*a;
    *a=*b;
    *b=t;
}
```

```
void main( )
{
    int x=10;
    int y=15;
    cout<<"x before swapping is:<<x<<"\n";
    cout<<"y before swapping is:<<y<<"\n";
    swap(&x,&y);
    cout<<"x after swapping is:<<x<<"\n";
    cout<<"y after swapping is:<<y<<"\n";
}
```