

Function:-

A Function, which can also be referred to as **subroutine**, **procedure**, **subprogram** or even **method**, carries out tasks defined by a sequence of statements called a **Statement Block** that need only be written once and called by a program as many times as needed to carry out the same task. Or may be defined as it is a group of statements that is executed when it is called from some point of the program.

Functions may depend on variables passed to them, called **Arguments**, and may pass results of a task on to the caller of the function, this is called the **Return Value**. The reasons to use functions are:-

- 1- To aid in the conceptual organization of a program.
- 2- To reduce program size.

We need to add three components to the C++ program, to result a function in the C++ program:-

- 1- Function declaration (prototype).
- 2- Calls to the function.
- 3- Function definition.

The structure of program will be as follows:-

```
# include <filename>
Function declaration;           // الاعلان عن الدالة
int main ()
{
.....
function calling;
.....                          // استدعاء الدالة
function calling;
}
function definition             // تعريف الدالة
{
// function body
}
```

Function Prototypes (declaration):-

The general form of the function declaration is:-

return-type function-name (argument list);

A function prototype tells the compiler the following information:-

- 1- The name of the function (This is simply a unique identifier).
- 2- The type of data returned by the function.
- 3- The number of parameters the function expects to receive.
- 4- The types of the parameters and the order in which these parameters are expected.

Many arguments may be passed to a function, while only one argument may be returned from it.

Examples:-

```
int max ( int, int ,int );
```

```
int max (int x, int y, int z);
```

The second statement same as the first.

```
float average (int , int );
```

```
void print( );
```

Calling Function:-

The function can be called many times in the program like:-

function-name (argument list);

Ex:- find the output of the following program:-

```
# include <iostream.h>
```

```
int square (int x) // here the function definition before main( ), so no prototype
```

```
{
```

```
int p;
```

```
p=x*x;
```

```
return p;}
```

```
int main( )
```

```
{
```

```
int i;
```

```
i=10;
```

```
cout<< square (i)<<" "; // first function call
```

```
i=square (12); // second function call
```

```
cout<<i<<" ";
```

```
i=5;
```

```
cout<<i<<" ";
```

```
cout<<square(i)<<" "<<"\n"; // third function call
```

```
return 0;}
```

Ex:- find the output of the following program:- (in this program we need prototype as follows):-

```
# include <iostream.h>
```

```
int square (int x) // here need prototype
```

```
int main( )
```

```
{
```

```
int i;
```

```
i=10;
```

```
cout<< square (i)<<" "; // first function call
```

```
i=square (12); // second function call
```

```
cout<<i<<" ";
```

```
i=5;
```

```
cout<<i<<" ";
```

```
cout<<square(i)<<" "<<"\n"; // third function call
```

```
return 0;}
```

```
int square (int x) // here the function definitions after main( ), so need prototype
```

```
{
```

```
int p;
p=x*x;
return p;}
```

Function Definition:-

From the examples above, we can see the actual code for the function is:-

return-type function-name (argument list)

```
{
// function body
.....
}
```

Important notes:-

- 1- The function don't end with ; semi comma.
- 2- The function is defined before and after main(), and it is compatible with the prototype
- 3- Function call means send the variable value to the function to be processed and return the result.
- 4- All variables declared in function definitions are local variables, they are known only in the function in which they are defined.
- 5- Function cannot defined inside another function.

Passing Arguments functions:-

There are three ways in C++ to pass arguments to a function:-

- 1- Call- by- value.
- 2- Call- by- reference with reference arguments.
- 3- Call- by-reference with pointer arguments.

- Call- by- value:-

Ex:- W.P in C++ to test the maximum number between two number?

Solution:-

```
#include<iostream.h>
int max (int ,int )
main( )
{
int m, n;
do {
cin>> m>> n;
cout<< max (m,n) << endl;
} while (m!= 0);
}
int max (int x, int y)
{
if (x<y) return y;
else return x;
}
```

Ex:- passing arguments to a function call by value that computes the square of an integer number?

Solution:-

```
#include<iostream.h>
int square (int);
int main( )
{
int number=5;
cout<<"The original value of number is"<< number;
square(number);
cout<<" The value of number after function call"<< number<<endl;
return 0;}
int square (int x)    // function definition
{
return x*x;}

```

Ex:- W.P in C++ to read the number and to print if the number is even or odd using function ?

Solution:-

```
#include<iostream.h>
void program(int );
int main( )
{
int number;
cout<< "Enter your number"<<endl;
cin>> number;
program(number);
return 0;}
void program (int x){
if (x%2==0)
cout<<"x="<<x<<,"\\t"<<"is even"<<"\\n";
else
cout<<"x="<<x<<,"\\t"<<"is odd"<<"\\n";
}

```

Ex:- W.P. to find the factorial of any number use the function?

Solution:-

```
#include<iostream.h>
int fact(int )
int main( ){
cout<<"Enter your number"<<endl;
int x;
cin>>x;
cout<<"factorial of"<<x<<"="<<fact(x)<<"\\n";
return 0;}

```

```
int fact (int y)
{
int x,i,p=1;
for (i=1;i<=y;i++)
p=p*i;
return p;}

```

Ex:- W.P in C++ that reads, sorts and prints an integer array a[5] using 3-function?

Solution:-

```
#include<iostream.h>
void setarray (int b[5]);
void sortarray (int b[5]);
void putarray (int b[5]);
int main( ){
int a[5];
setarray (a);
sortarray (a);
cout<<"sorted array is";
putarray(a);}
void setarray( int b[5])
{
cout<<"Enter five integer array: ";
for(int i=0; i<5; i++)
cin>> b[i];
}
void sortarray (int b[5])
{
for ( int i=0; i<4;i++)
for ( int j=i+1; j<5; j++)
if (b[i] > b[j])
{
int temp = b[i];
b[i]= b[j];
b[j] = temp;
}
}
void putarray( int b[5])
{
for (int j=0;j<5; j++)
cout<< b[j] <<" ";
cout<<endl;
}

```

Passing by Reference:-

The symbol & is used to declare a reference:-

Ex:- passing arguments to a function call by reference?

Solution:-

```
#include<iostream.h>
void square (int &);
int main( )
{
    int number=5;
    cout<<"The original value of number is"<< number;
    square(number);
    cout<<" The value of number after function call"<< number<<endl;
    return 0;}
void square (int &y)    // function definition
{
    return y=y*y;}
```

Ex:- W.P in C++ that reads 2 integers and use function to swap their values using call by reference?

Solution:-

```
#include<iostream.h>
void swap( int &,int &);
int main( )
{
    int a,b;
    cout<<" enter the two numbers:";
    cout<<"a=";
    cin>>a;
    cout<<"b=";
    cin>>b;
    swap(a,b);
    cout<<" after swap process"<<endl;
    cout<<"a="<<a<<" "<<"b="<<b<<endl;
    return 0;}
void swap( int&x, int&y)
{
    int w=x;
    x=y;
    y=w;}
```

Ex:- W.P. in C++ to find the summation, subtraction, and multiplication of any two numbers using function (calling by reference)?

Solution:-

```
#include<iostream.h>
void calculate( int, int, int&, int&, int&);
int main( ){
int a,b,sum,subtract,multi;
cout<<"enter two integer numbers"<<endl;
cin>>a>>b;
calculate(a,b, sum,subtract,multi);
cout<<" the sum="<<sum;
cout<<"\n the subtract="<< subtract;
cout<<"\n the multi="<<multi;
return 0;}
void calculate( int a, int b, int &sum, int &subtract, int &multi);
{
sum=a+b;
subtract=a-b;
multi=a*b;}
```

Important notes:-

- 1- When we need the function returns more than one value, we can pass arguments by reference to the function.
- 2- A reference provides an alias-another name- for a variable.
- 3- The passing by reference is that the function can access the actual variable in the calling program. This provides a mechanism for passing more than one value from the function back to the calling program.