User defined Functions –

- In programming, function refers to a segment that groups code to perform a specific task. Depending on whether a function is predefined or created by programmer; there are two types of function:

  1. **Library Function**
  2. **User-defined Function**

- Library Function – Library functions are the built-in function in C++ programming. Programmer can use library function by invoking function directly; they don't need to write it themselves.

- User defined Function - C++ allows programmer to define their own function. A user-defined function groups code to perform a specific task and that group of code is given a name (identifier). When the function is invoked from any part of program, it all executes the codes defined in the body of function.

Example 1: Library Function -

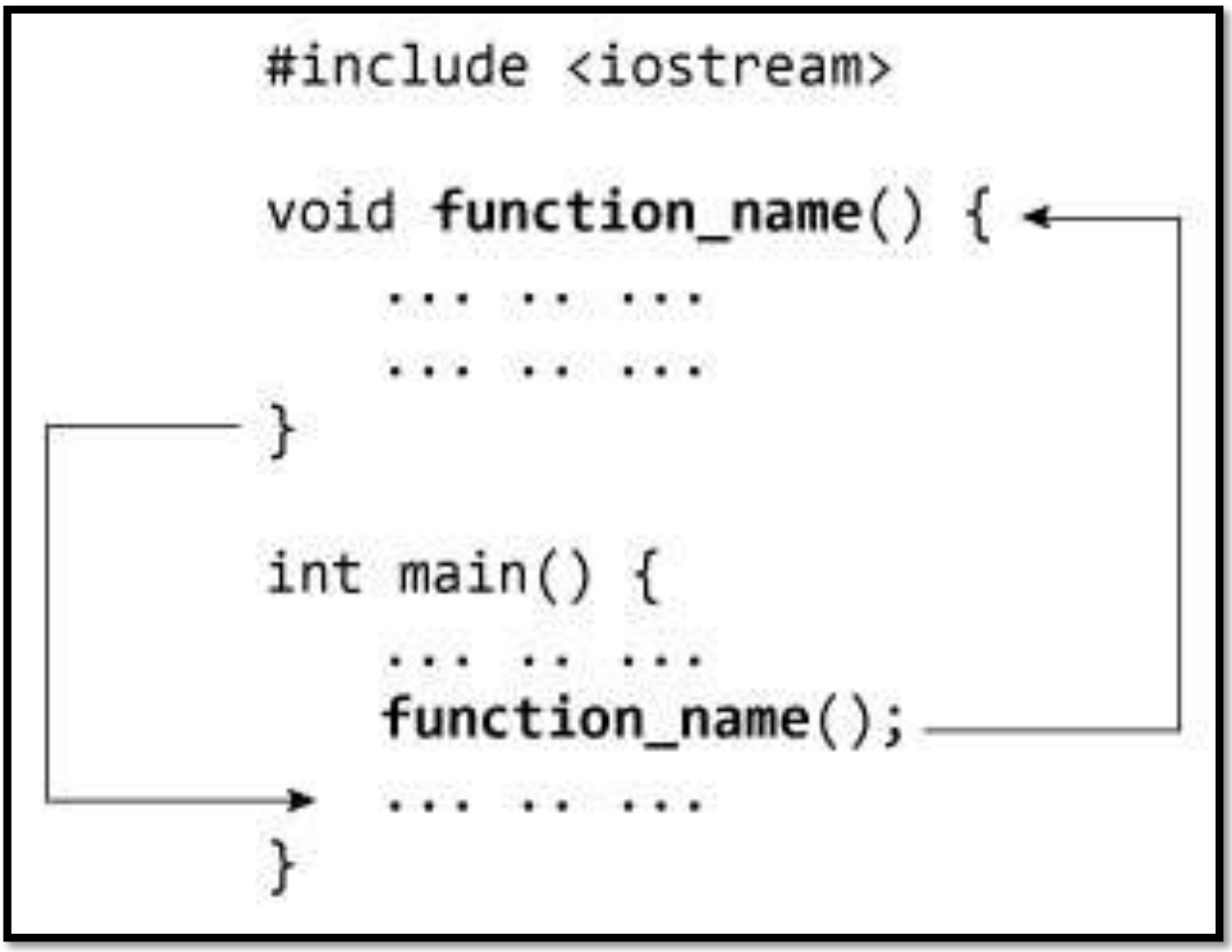
```
#include<iostream.h>
#include<cmath.h>

void main()
{
   double number, squareRoot;
   cout << "Enter a number: ";
   cin >> number;

   squareRoot = sqrt(number);
   cout << "Square root of " << number << " = " << squareRoot;

}
```

How user-defined function works in C Programming?



Example 2: User Defined Function -

```
#include<iostream.h>

// Function prototype (declaration)
int add(int, int);

void main()
{
   int num1, num2, sum;
   cout<<"Enters two numbers to add: ";
   cin >> num1 >> num2;

   // Function call
   sum = add(num1, num2);
   cout << "Sum = " << sum;
}
```

```
// Function definition
int add(int a, int b)
{
   int add;
   add = a + b;
   // Return statement
   return add;
}
```

Passing Arguments to Function -

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

int add(int, int);

int main() {
    ... .. ...
    sum = add(num1, num2);   // Actual parameters: num1 and num2
   ... .. ...
}

int add(int a, int b) {      // Formal parameters: a and b
    ... .. ...
     add = a+b;
   ... .. ...
}
```

Different types of user defined functions -

- ❑ *With out return type, with out parameters*
- ❑ *With out return type, with parameters*
- ❑ *With return type, with out parameters*
- ❑ *With return type, with parameters*

```
/* Type - 1: Without return type, without parameters */
#include<iostream.h>
#include<conio.h>

//function declaration
void add();

void main()
{
	//function call
	add();
}

//function body
void add()
{
	int a,b,c;
	cout<<"Enter the numbers: ";
	cin>>a>>b;

	c = (a + b);
	cout<<"Sum is: "<<c;
}

/* Type - II: Without return type, with parameters - Call by Value/Pass by Value */
#include<iostream.h>
#include<conio.h>

//function declaration
void add(int p,int q);

void main()
{
	int a,b;
	cout<<"Enter the numbers: ";
	cin>>a>>b;
```

```
        //function call
        add(a,b);  //a,b = Actual parameters (Arguments)
}

//function body
void add(int p,int q)  //p,q = Formal parameters
{
        int c;

        c = (p + q);
        cout<<"Sum is: "<<c;
}
```

****