Functions

What is Function?

- A function is a block of code that performs a specific task.
- A function executes only when it is called.
- You can pass data, known as parameters, into a function.
- Functions are used to perform certain actions, and they are important for reusing code: Define the code once, and use it many times.

Types of function

- There are two types of function in C programming:

1. Standard library functions (Pre-Defined Functions)
2. User-defined functions

Standard library functions (Pre-Defined Functions)

- The standard library functions are built-in functions in C programming.
- These functions are defined in header files. For example,

The printf() is a standard library function to display formatted output to the screen. This function is defined in the stdio.h header file. Hence to use printf() function, we need to include the stdio.h header file using #include <stdio.h>.

The sqrt() function calculates the square root of a number. The function is defined in the math.h header file.

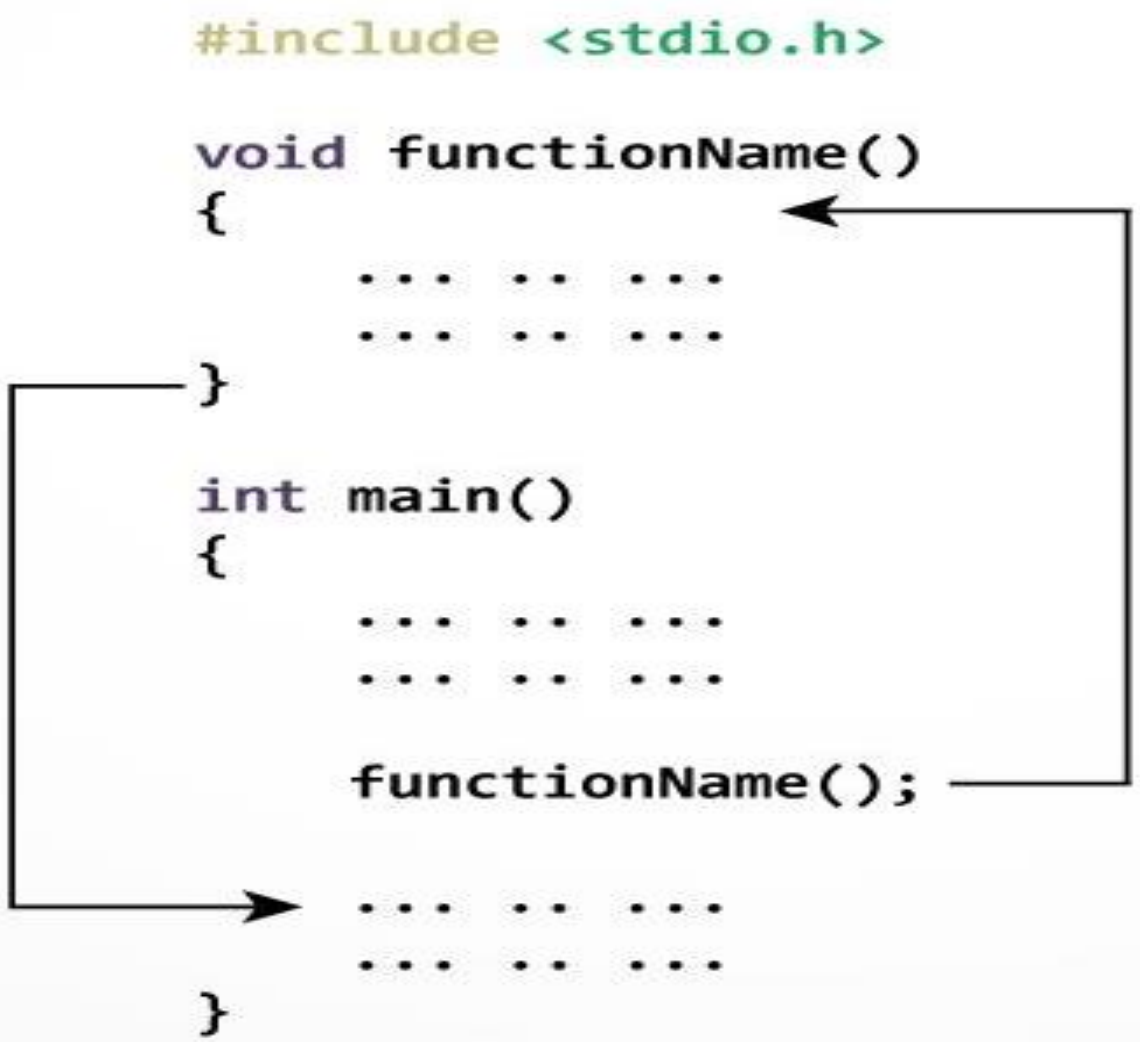
User-defined functions

- User-defined functions are **a block of code written by the user to perform a specific action**.
- A user-defined function has a return type, a function name, parameters, and body of the function.
- Function can be called using the unique name of the function followed by function parameters passed inside round brackets ().

Advantages of user-defined function

- The program will be easier to understand, maintain and debug.
- Reusable codes that can be used in other programs
- A large program can be divided into smaller modules. Hence, a large project can be divided among many programmers.

How function works in C programming?



How Function works

- The execution of a C program begins from the main() function
- When the compiler encounters functionname() control of the program jumps to void functionname()
- And, the compiler starts executing the codes inside functionname().
- The control of the program jumps back to the main() function once code inside the function definition is executed.

Example of user-defined function with arguments and with return value

```
#include <stdio.h>
int addNumbers(int a, int b);       // function declaration

int addNumbers(int a, int b)       // function definition
{
   int result;
   result = a+b;
   return result;              // return statement
}

void main()
{
   int n1,n2,sum;
   printf("Enters two numbers: ");
   scanf("%d %d",&n1,&n2);

   sum = addNumbers(n1, n2);       // function call
   printf("sum = %d",sum);
}
```

Example of user-defined function with no arguments & no return value

```
#include <stdio.h>
Hello();        // function declaration

Hello()         // function definition
{
   printf("Hello Welcome");
}

void main()
{
   Hello();                 // function call
}
```

Example of user-defined function with no arguments and a return value.

```
#include <stdio.h>
int addNumbers();       // function declaration

int addNumbers()       // function definition
{
   int n1,n2,result;
   printf("Enter two numbers: ");
   scanf("%d %d",&n1,&n2);
   result = n1+n2;
   return result;              // return statement
}

void main()
{
   int sum;
  sum = addNumbers();       // function call
   printf("sum = %d",sum);
}
```

Example of user-defined function with arguments but no return value.

```
#include <stdio.h>
addNumbers(int a, int b);       // function declaration

addNumbers(int a, int b)       // function definition
{
   int result;
   result = a+b;
    printf("Sum = %d",result);            // return statement
}

void main()
{
   int n1,n2;
   printf("Enters two numbers: ");
   scanf("%d %d",&n1,&n2);
  addNumbers(n1, n2);      // function call
}
```