

Functions

- Function is a block of code used to perform a specific task
- Every C program has at least one function `main()`.
- Functions can call other functions and/or itself.
- Functions improve reusability and readability.
 - Using functions, large programs can be divided into smaller modules
 - Functions can be called any number of times

Function Types

- Standard library functions
 - Examples: `printf()`, `scanf()`, `pow()`, `sqrt()`, ...
 - Declared in header files (`stdio.h`, `math.h`, ...)
- User defined functions
 - Created by the user

Requirements

- Function declaration (function prototype)
- Function definition
- Function call

Declaration/Prototype

- Contains the block of code required to perform a specific task
- Consists of a function header and a function body

Syntax:

```
returnType functionName (parameterList) ;
```


Definition

- Contains the block of code required to perform a specific task
- Consists of a function header and a function body

Syntax:

```
returnType functionName (parameterList)
{
    // body statements & return statement
}
```

Function Call

- The control of the program is transferred to the function definition.
- The function name must match exactly the name of the function in the function prototype and function header.
- The number and data type of parameters “passed” to the function must match the number and data type of parameters from the function prototype’s parameter list and function header’s parameter list.
- If a function returns a value, the function call must be assigned to a variable in order to capture returned value. The data type of a variable used to store returned value must match the function return type.