_____

Learning Outcomes:
- Students should be able to think logically and develop problem-solving skills

- Students should be able to define and use functions

- Students should be able to break their problem into small reusable functions.

Introduction

Each C++ program has at least one function, the main() function. You can divide your code into different functions. This division should be such that every function does a specific task. Now, we need to make reusable functions which are also called procedures or sub-programs.
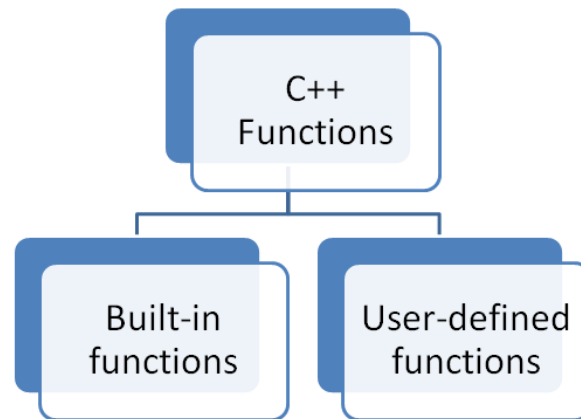
Functions

Take a look from your slides. A function is a block of instructions that is executed when it is called from some other point of the program. The following picture explains the different components of the **Function**.
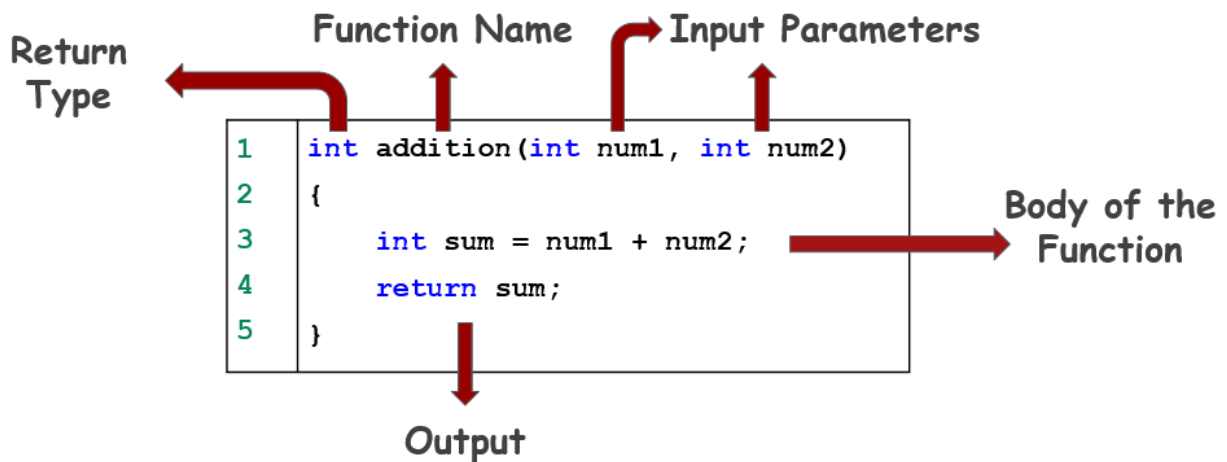
There are two types of function:

1. **Built-in Functions:** Predefined in C++
2. **User-defined Functions:** Created by users

Built-in Functions:

Programmers can use built-in functions by invoking the functions directly; they don't need to write the functions themselves. Some common built-in functions in C++ are sqrt(), sabs(), isdigit(), etc. In order to use library functions, we usually need to include the header file in which these library functions are defined. For instance, in order to use mathematical functions such as sqrt() and fabs(), we need to include the header file **cmath**.

Example 1:
Write a C++ Program to Find the Square Root of a Number using a pre-defined sqrt() function.

Solution
| ```cpp
#include <iostream>
#include <cmath>
using namespace std;
main()
{
    double number, squareRoot;
    number = 25.0;
    /* sqrt() is a library function to calculate the square root */
    squareRoot = sqrt(number);
    cout << "Square root of " << number << " = " << squareRoot;
}
``` |
| **Output** |
| ```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
Square root of 25 = 5
``` |

Example 2:

Write a C++ Program to Find the square of a Number using a pre-defined pow() function.

Solution

```cpp
#include <iostream>
#include <cmath>
using namespace std;
main()
{
    double number, result;
    number = 25.0;
    // pow() is a library function to calculate the power
    result = pow(number, 2);
    cout << "Square of " << number << " = " << result;
}
```

Output

```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
Square of 25 = 625
```

Example 3:
Write a C++ Program to Find the sin of a Number using a pre-defined sin() function.

Note: The sin() function in C++ returns the sin of an angle (argument) given in radians. This function is as defined in <cmath> header file

Solution

```cpp
#include <iostream>
#include <cmath>
using namespace std;
main()
{
    double number, result;
    number = 5.0;
    // sin() is a library function to calculate the sin
    result = sin(number);
    cout << "Sin of " << number << " = " << result;
}
```
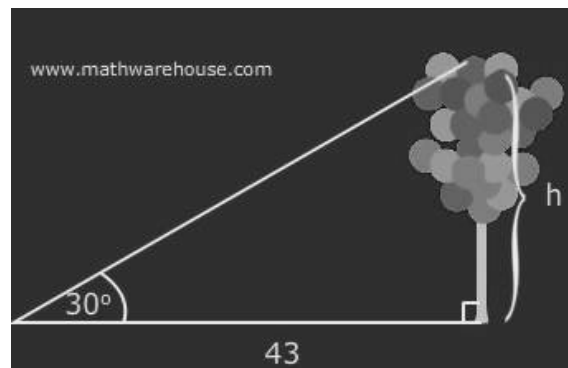
Output

```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
Sin of 5 = -0.958924
```

Challenge 1:
The angle of elevation from a point 43 feet from the base of a tree on level ground to the top of the tree is 30° (30 degree). Write a C++ program to calculate the height of the tree?

Hint:
Use pre-defined sin(), cos() and tan() functions which are defined in cmath library.
Remember: 1 radian = 57.2958 degrees

Output:
Correct Answer is : h = 24.8261

Challenge 2:
Write a C++ program that calculates the value of x for the following equation
$$5x^2 + 6x + 1 = 0$$
Hint:
Just put the values of a, b and c into the Quadratic Formula, and do the calculations.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Output:
Answer: x = −0.2 or x = −1

**Congratulations!!! You have become an expert in using pre-defined Functions.**
Lets become an expert in user defined functions now.

User-defined Functions:

C++ allows the 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 the program, it all executes the codes defined in the body of the function.

We have studied 2 types of user-defined functions.

1. Value Returning Function
2. Void Function (which returns nothing)
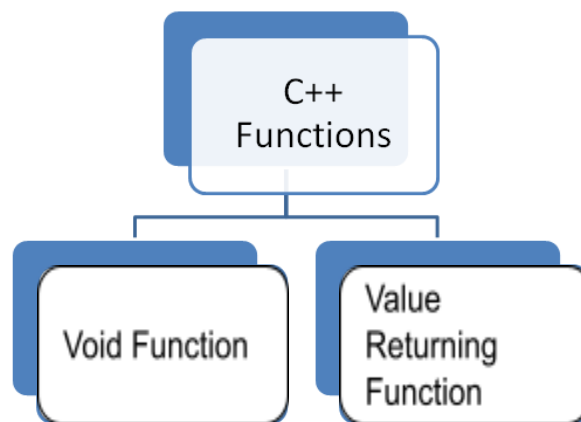
The General syntax to declare a function is:

```cpp
returnType functionName(dataType parameter1, dataType parameter2,...) {
    // function body
}
```

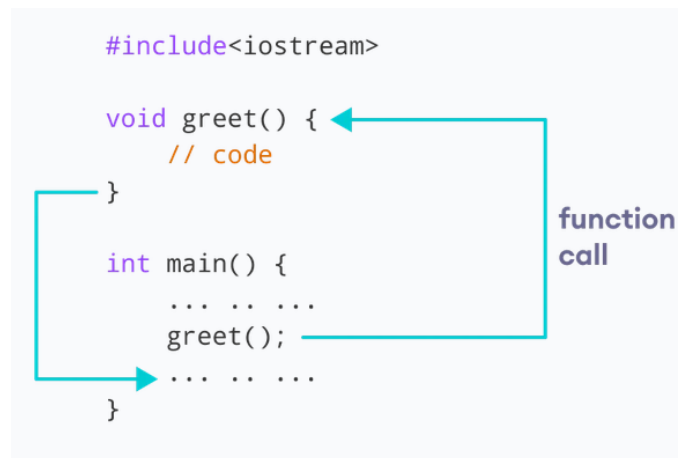## Void Functions:

Example 4:

Here's an example of a void function (which returns nothing) declaration. Suppose we want to make a void function that only displays "Hello World" on the console

```cpp
// function declaration
void greet() {
    cout << "Hello World";
}
```

We have declared a function named greet(). To use the greet() function, we need to call it.
Here's how we can call the above greet() function.

```cpp
#include <iostream>
using namespace std;
main()
{
    // calling a function
    greet();
}
```

Following Figure shows how this function works.

```cpp
#include<iostream>

void greet() {
    // code
}

                                    function
                                    call
int main() {
    ... .. ...
    greet();
    ... .. ...
}
```

Solution 1: (Writing the function before main)
| ```cpp
#include <iostream>
using namespace std;

// function declaration
void greet() {
    cout << "Hello World";
}

main()
{
    // calling a function
    greet();
}
``` |
| **Output** |

```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
Hello World
```

Solution 2: (What if we write the function after the main)

```cpp
#include <iostream>
using namespace std;
main()
{
    // calling a function
    greet();
}

// function declaration
void greet() {
    cout << "Hello World";
}
```

Output (Error)

```
C:\C++\Week6>c++ 5.cpp -o 5.exe
5.cpp: In function 'int main()':
5.cpp:8:5: error: 'greet' was not declared in this scope
    8 |     greet();
      |     ^~~~~
```

Function Prototype:

In C++, the code of function declaration should be before the function call. However, if we want to define a function after the function call, we need to use the function prototype.

Ideal Solution: (Writing the function after main but writing the function prototype before main)

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

// function prototype
void greet();
```

```
main()
{
    // calling a function
    greet();
}

// function declaration
void greet() {
    cout << "Hello World";
}
```

Output

```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
Hello World
```

In the last example, there was no parameter passed to the function. Lets see a void function which has 2 parameters.

Example 5: Write a program to print 2 numbers, one of int type and one of float type.

Solution

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

//  Function prototype
void displayNum(int n1, float n2);

main() {
    int num1 = 5;
     float num2 = 5.5;

    // calling the function
    displayNum(num1, num2);

}

// Function definition
void displayNum(int n1, float n2) {
    cout << "The int number is " << n1 << endl;
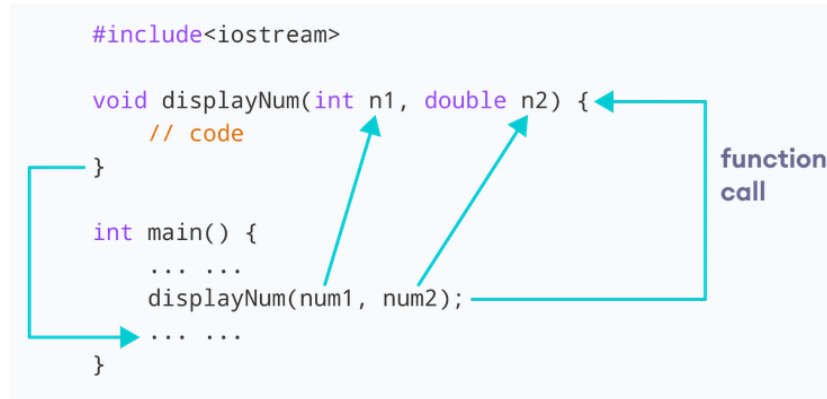    cout << "The float number is " << n2;
}
```

Output

```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
The int number is 5
The float number is 5.5
```

Following figure shows how this works


```
#include<iostream>

void displayNum(int n1, double n2) {
    // code
}                                          function
                                           call
int main() {
    ... ...
    displayNum(num1, num2);
    ... ...
}
```

(Note: in the figure code they have written the function before main and they have not written the prototype of the function. Figure is just to show you how the function call is made. for your solution the function is written after the main.)

Value Returning Functions:

In the above programs, we have used void in the function declaration. It's also possible to return a value from a function. For this, we need to specify the returnType of the function during function declaration. Then, the return statement can be used to return a value from a function. Syntax is as follows:

```
returnType functionName(dataType parameter1, dataType parameter2,...) {
    // function body
    return value;
}
```

Example 6: Write a function to add two numbers.

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

```
// function prototype
int add(int a, int b);

main() {
    int sum;
    // calling the function and storing
    // the returned value in sum
    sum = add(100, 78);
    cout << "100 + 78 = " << sum << endl;
}

// declaring a function
int add(int a, int b) {
    return (a + b);
}
```

Output

```
C:\C++\Week6>c++ 5.cpp -o 5.exe

C:\C++\Week6>5.exe
100 + 78 = 178
```

Congratulations!!! You have learnt about the user-defined functions as well.
Now, let's start the challenges to become an expert in user-defined functions and from now on we will always write our code in the form of functions (because this is a good practice and all the great programmers write their code in small reusable functions).

Challenge 3:

Create a function that takes two numbers and a mathematical operator + - / * and will perform a calculation with the given numbers and returns the output result.

Test Cases:

calculator(2, "+", 2) → 4
calculator(2, "*", 2) → 4
calculator(4, "/", 2) → 2

Challenge 4:
A person is eligible to vote if his/her age is greater than or equal to 18. Define a function to find out if he/she is eligible to vote.

Challenge 5:

Write two functions to print the maximum and the minimum number respectively among four numbers entered by the user.

Challenge 6:

Write a function for checking whether the alphabet entered by the user is in small case or in capital case (Suppose user can only enter 'A' or 'a'). Make a function that takes 1 Character as input, does processing according to the input and then returns the string. String is "You have entered Capital A" if the user enters 'A', otherwise "You have entered small A".

Challenge 7:

You've been hired by Automobile company to write a program to help the tax collector calculate vehicle taxes. Vehicle taxes are based on two pieces of information; the price of the vehicle and the vehicle type code.

$$Tax\ Amount = Item\ Price\ *\ \frac{TaxRate}{100}$$

The formula of calculating the final price of an item is:

Final Price = Item Price + Tax Amount

Tax rates are in the table below.

Vehicle Type	Vehicle Code	Tax Rate
Motorcycle	M	6%
Electric	E	8%
Sedan	S	10%
Van	V	12%
Truck	T	15%

After the tax has been calculated, the program should display the following on the screen;
The final price on a vehicle of type xxx after adding the tax is $xxx.
with xxx replaced by the vehicle type and $xxx with the final price.

Your job is to write a function
float taxCalculator(char type, float price);
and then write the main function for taking the input from the user and then displaying the final output.