Defining a Function

In this lesson, we will see how to define our own function in C++.

We'll cover the following

Function definition

A function's definition tells what our function will do when it is called. The basic syntax for defining a function in C++ is:

```
return_type  function_name  ( function_parameters )
{
function_body

}
```

We have already discussed the return_type, function_name, and function_parameters in the previous lesson. Let's discuss the function_body.

function_body

A function body consists of a group of statements that do a particular task. We write our function code inside the curly braces. Everything written inside the curly braces is what the function does when it is called.

Main function

In the code below, you see the highlighted lines in every C++ program. If you look closely at these lines, you see that the `main( )` is the function here. It is the point

from where every C++ program starts its execution. Whenever the C++ program is executed, the operating system gives control to the `main` function.

> 📝 Every program in C++ must have a `main` function.

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

int main() {
  // your code goes here

  return 0;
}
```

Anatomy of the `main` function

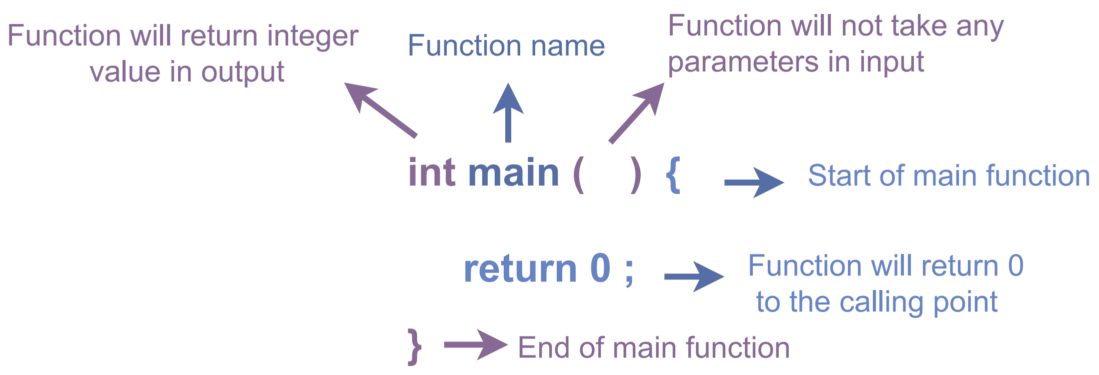
`int` specifies that the `main` function returns an integer value in the output.

`{` indicates the beginning of the `main` function.

`return 0` returns 0 to the calling point on the successful execution of the program.

> 📝 **Note:* Adding a return 0 statement in a program is not mandatory.

`}` indicates the end of the `main` function.


Function will return integer value in output
Function name
Function will not take any parameters in input
int main ( ) {
Start of main function
return 0 ;
Function will return 0 to the calling point
}
End of main function

Example program

Consider the blender example given in this lesson. Let's declare and define a function `make_juice`.

```
#include <iostream>

using namespace std;
// Function declaration
int make_juice(int water, int fruit);

int main() {

  return 0;

}

// Function definition
int make_juice(int water, int fruit) {
  // Define new  variable juice of int type
  int juice;
  // Adds water in apple and save output in juice
  juice = water + fruit;
  // Prints text on the screen
  cout << "Your juice is ready" << endl;
  // Returns juice value in output
  return juice;

}
```

Explanation

In the code above:

Line No. 5: Declares the function `make_juice`

Lines No. 14 to 24: Defines function `make_juice`

Line No. 14: `make_juice` is the name of the function. It takes the number of glasses of `water` and the number of `fruits` as input parameters. The function returns the number of `juice` glasses in the output.

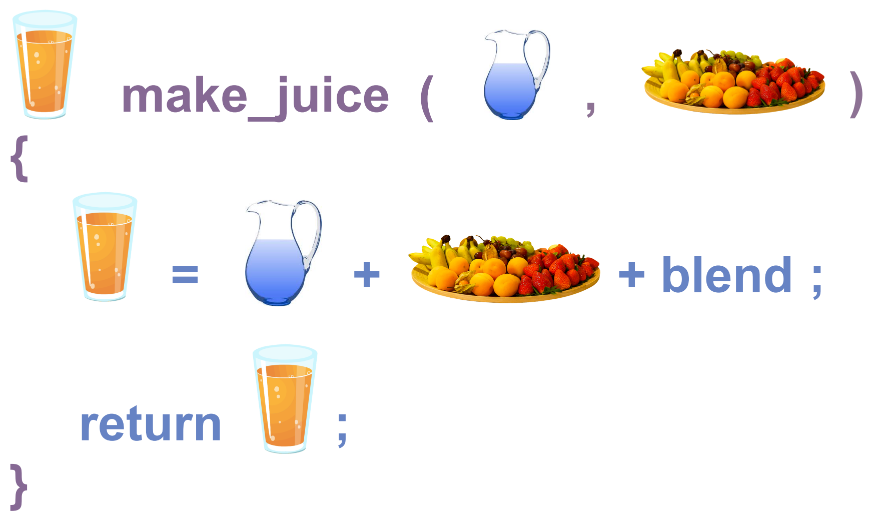
Line No. 16: Declares a variable `juice`

Line No. 18: Adds `water` in the `fruit` and saves the output in `juice`

Line No. 20: Prints `Your juice is ready` to the console

Line No. 22: Returns the number of glasses of `juice` in the output (Adding more fruit and water in the input returns a greater number of juice glasses in the output)

output).



Quiz

Q

Define a function `number_sum` that takes the `num1` and `num2` in the input and returns their sum in output. `num1` and `num2` take integer values.

(You can select multiple correct answers)

Retake Quiz

That’s all about defining a function. Let’s see how to call our function in a program.

See you there!