Parameters

This lesson explains how the parameters passed in the main function are assigned and used in the function itself using an example.

We'll cover the following

Parameters Syntax

Parameters are how *data* is passed between functions through the *call* of the function.

We learned earlier that we list the data we want to pass to a function in the call between the `( )`. The order of the list is determined by the function *definition*. The **first** parameter in the list will be *assigned* to the *variable* listed **first** in the function *definition*.

> **Note:** In most cases, you must have the correct number and type of data being passed to the function or you will receive an error when you try to compile your program.

Example

Let's take a look at an example below:

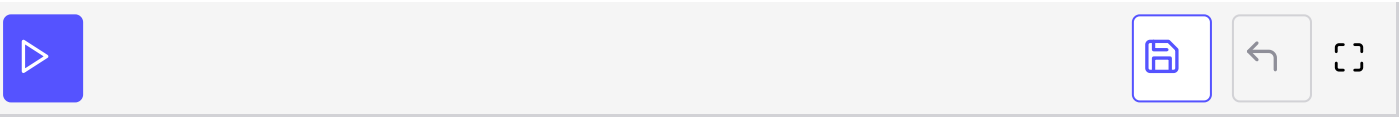
```
#include <iostream>
using namespace std;

int fctn(int arg1, int arg2);

int main()
{
   int answer;
   int num1 = 10;
```

```
   answer = fctn(num1, 12);    // num1 and 12 are arguments passed to fctn
   cout << "Answer is: " << answer << endl;
}


int fctn(int arg1, int arg2)    // function definition
{
  cout << "num1 is assigned to arg1, value of arg1 is: "<<arg1<<endl;
  cout << "12 is assigned to arg2, value of arg2 is: "<<arg2<<endl;
  return arg1*arg2;      // mutliplying arg1 and arg2 and returning the answer
}
```



Explanation

As you can see above, the contents of the variable `num1` are passed to `arg1` and the *integer* value `12` is passed to `arg2`. The function `fctn` then returns the product after performing *multiplication* operation on `arg1` and `arg2`.

In the next lesson, we will further discuss the *methods* used to pass *parameters* to the functions.