Function Parameters & References

In C++, function parameters are variables that are used to pass data into a function when it is called. Function parameters are defined in the function's declaration and are used to receive the data that is passed to the function when it is called.

Function parameters can be either pass-by-value or pass-by-reference. Pass-by-value means that the function receives a copy of the data that is passed to it, while pass-by-reference means that the function receives a reference to the original data.

Here is an example of a function that takes two integer parameters and returns their sum, using pass-by-value:

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

// function that takes two integers as input and returns thei
int add(int x, int y) {
  return x + y;
}

int main() {
  int a = 3;
  int b = 4;
  int c = add(a, b); // c will be assigned the value 7
  cout << c << endl;
  return 0;
}
```

Output

```
7
```

In this example, the function **add()** takes two integer parameters, **x** and **y**, and returns their sum. The function is called from the **main()** function with the values **a** and **b**, and the result is assigned to the variable **c**.

To pass a parameter by reference, you can use the **&** operator in the function declaration to indicate that the function should receive a reference to the original data. Here is an example of the same function using pass-by-reference:

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

// function that takes two integers as input and returns thei
int add(int &x, int &y) {
  return x + y;
}

int main() {
  int a = 3;
  int b = 4;
  int c = add(a, b); // c will be assigned the value 7
  cout << c << endl;
  return 0;
}
```

Output

```
7
```

In this example, the function **add()** takes two integer references, **x** and **y**, and returns their sum. The function is called from the **main()** function with the variables **a** and **b**, and the result is assigned to the variable **c**.

Passing parameters by reference can be more efficient than pass-by-value because it avoids the overhead of creating and copying the data. However, it is important to be careful when using pass-by-reference, as it can lead to unintended side effects if the function modifies the original data.

When you pass a parameter by reference to a function, you are essentially giving the function a direct reference to the original data. This means that the function can modify the original data directly, rather than working with a copy of the data.

While passing parameters by reference can be more efficient, it can also lead to unintended side effects if the function modifies the original data.

To avoid unintended side effects when using pass-by-reference, it is important to carefully consider whether the function needs to modify the original data or if it can work with a copy of the data. In general, pass-by-reference should only be used if the function needs to modify the original data, as it can lead to unintended consequences if the function modifies the data in unexpected ways.