**Overloaded functions.**

In C++, two different functions can have the same name if their parameters are different; either because they have a different number of parameters, or because any of their parameters are of a different type.

For example:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 | // overloading functions  #include <iostream>  using namespace std;  int operate (int a, int b)  {  return (a\*b);  }  double operate (double a, double b)  {  return (a/b);  }  int main ()  {  int x=5, y=2;  double n=5.0, m=2.0;  cout << operate (x,y) << '\n';  cout << operate (n,m) << '\n';  return 0;  } | 10  2.5 | [Edit & Run](https://cplusplus.com/doc/tutorial/functions2/) |

In this example, there are two functions called *operate*, but one of them has two parameters of type *int*, while the other has them of type *double*. The compiler knows which one to call in each case by examining the types passed as arguments when the function is called. If it is called with two *int* arguments, it calls to the function that has two *int* parameters, and if it is called with two *doubles*, it calls the one with two *doubles*.

In this example, both functions have quite different behaviours, the int version multiplies its arguments, while the double version divides them. This is generally not a good idea. Two functions with the same name are generally expected to have -at least- a similar behaviour, but this example demonstrates that is entirely possible for them not to. Two overloaded functions (i.e., two functions with the same name) have entirely different definitions; they are, for all purposes, different functions, that only happen to have the same name.

Note that a function cannot be overloaded only by its return type. At least one of its parameters must have a different type. Or it should have different number of parameters.

Aforementioned statements imply and testify that the below shown program will not get compiled:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 | // error when overloading functions  #include <iostream>  using namespace std;  int operate(double a, double b)  {  return (a \* b);  }  double operate(double a, double b)  {  return (a / b);  }  int main()  {  int x = 5, y = 2;  double n = 5.0, m = 2.0;  cout << operate(x, y) << '\n';  cout << operate(n, m) << '\n';  return 0;  } | Error: unable to run because cannot overload functions that are distinguished only by the type of return value | [Edit & Run](https://cplusplus.com/doc/tutorial/functions2/) |

But it will work when it is like this:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22  23  24  25  26  27  28  29  30 | // valid overloading functions  #include <iostream>  using namespace std;  double operate(double a, float b)  {  return (a - b); }  double operate(double a)  {  return (a \* 8);  }  double operate(double a, double b)  {  return (a / b);  }  int operate(float a, double b)  {  return (a + b);  }  int main()  {  int x = 8;  float y = 8;  double n = 5.0, m = 2.0;  cout << operate(n, y) << '\n';  cout << operate(x) << '\n';  cout << operate(n, m) << '\n';  cout << operate(y, m) << '\n';  return 0;  } | -3  64  2.5  10 | [Edit & Run](https://cplusplus.com/doc/tutorial/functions2/) |