



Adapted from C++ How To Program edited for our own purposes

Function Overloading

C++ enables several functions of the same name to be defined, as long as they have different signatures. This is called function overloading. The C++ compiler selects the proper function to call by examining the number, types and order of the arguments in the call.

Function overloading is used to create several functions of the same name that perform similar tasks, but on different data types. For example, many functions in the math library are overloaded for different numeric types—the C++ standard requires float, double and long double overloaded versions of the math library functions

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

// function square for int values
int square( int x )
{
    cout << "square of integer " << x << " is ";
    return x * x;
} // end function square with int argument

// function square for double values
double square( double y )
{
    cout << "square of double " << y << " is ";
    return y * y;
} // end function square with double argument

int main()
{
    cout << square( 7 ); // calls int version
    cout << endl;
    cout << square( 7.5 ); // calls double version
    cout << endl;
} // end main
```

```
square of integer 7 is 49
square of double 7.5 is 56.25
```