# Function Overloading

Function overloading is the process of using the same name for two or more functions. The secret to overloading is that each redefinition of the function must use either different types of parameters or a different number of parameters. It is only through these differences that the compiler knows which function to call in any given situation. For example, this program overloads myfunc( ) by using different types of parameters.

//program overloads myfunc( ) by using different types of parameters.//

#include <iostream>

using namespace std;

int myfunc(int i); // these differ in types of parameters

double myfunc(double i);

int main()

{

cout << myfunc(10) << " "; // calls myfunc(int i)

cout << myfunc(5.4); // calls myfunc(double i)

return 0;

}

double myfunc(double i)

{

return i;

}

int myfunc(int i)

{

return i;

}

//program overloads myfunc( ) using a different number of parameters://

#include <iostream>

using namespace std;

int myfunc(int i); // these differ in number of parameters

int myfunc(int i, int j);

int main()

{

cout << myfunc(10) << " "; // calls myfunc(int i)

cout << myfunc(4, 5); // calls myfunc(int i, int j)

return 0;

}

int myfunc(int i)

{

return i;

}

int myfunc(int i, int j)

{

return i\*j;

}

