OOPS

Function Overloading

After classes, perhaps the next most important feature of C++ is function overloading.

Two or more functions can share the ***same name*** as long as ***either the type of their arguments differs or the number of their arguments differs - or both***.

When two or more functions share the same name, they are said ***overloaded***.

Overloaded functions can help reduce the complexity of a program by allowing related operations to be referred to by the same name.

To overload a function, simply declare and define all required versions. ***The compiler will automatically select the correct version based upon the number and/or type of the arguments*** used to call the function. The following example illustrates the overloading of the absolute value function:

```
#include <iostream>
using namespace std;
// overload myabs three ways
int myabs(int n);
long myabs(long n);
double myabs(double n);
int main() {
   cout << "Abs value of -10: " << myabs(-10) << "\n";
   cout << "Abs value of -10L: " << myabs(-10L) << "\n";
   cout << "Abs value of -10.01:" << myabs(-10.01) << "\n";
   return 0;
}

// myabs() for ints
int myabs(int n) {
   cout << "In integer myabs()\n";
   return n < 0 ? -n : n;
}

// myabs() for long
long myabs(long n) {
   cout << "In long myabs()\n";
   return n < 0 ? -n : n;
}


// myabs() for double
double myabs(double n) {
   cout << "In double myabs()\n";
   return n < 0 ? -n : n;
}
```

The compiler automatically calls the correct version of the function based upon the type of data used as an argument.

Overloaded functions can also differ in the number of arguments. But, you must ***remember that the return type alone is not sufficient to allow function overloading***. If two functions differ only in the type of data they return, the compiler will not always be able to select the proper one to call. For example, the following fragment is ***incorrect***,

```
// This is incorrect and will not compile
int f1 (int a);
double f1 (int a);
f1(10); // which function does the compiler call???
```

NO NOTES TO SHOW