Passing by Pointer and Reference

Normally, when function is called, copies of the arguments are made. As a result, changing the values of a functions parameters doesn't effect the code that called the function. Here is an example, notice that the function changing the value of size does not change deskHeight.

outputSize(int size) {

cout << size << " inches" << endl;

size = 20;

}

int main() {

double deskHeight = 30;

outputSize(deskHeight);

// deskHeight is still 30 here

return 0;

}

 But now we know about pointers, what if we passed a pointer to a function?

outputSize(int \*size) {

cout << \*size << " inches" << endl;

\*size = 20;

}

int main() {

double deskHeight = 30;

outputSize(&deskHeight);

// deskHeight is now 20!!!!!

return 0;

}

Dealing directly with pointers can be dangerous, so there is a safer alternative called passing by referece. All we have to do is put an & in front of a paramter and it will be passed by referece. Notice that the behavior is the same as passing by pointer, but we don't have to dereferece the parameter.

outputSize(int &size) {

cout << size << " inches" << endl;

size = 20;

}

int main() {

double deskHeight = 30;

outputSize(deskHeight);

// deskHeight is now 20!!!!!

return 0;

}