NOTES

PARAMETER TRANSMISSION

Parameters to a function may be passed in two different ways. The following examples illustrate these two different methods.

CALL BY VALUE - a copy of the actual argument is placed in the parameter
In effect, the parameter is like a local variable with an initial Value. Any change to the parameter does NOT effect the calling program.

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

void change_to_five(int);

int main () {
   int val = 66;

     cout << "val is" << val << endl;
     change_to_five(val);
     cout << "val is" << val << endl;

     return 0;
}

void change_to_five(int number){

       number = 5;
}
```

Output: val is 66
val is 66

CALL BY REFERENCE- any reference to the parameter uses the actual argument itself. Thus any change made to the parameter variable is really changing the argument location.

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

void change_to_five(int&);

int main () {
   int val = 66;

    cout << "val is" << val << endl;
    change_to_five(val);
    cout << "val is" << val << endl;

    return 0;
}

void change_to_five(int& number){

        number = 5;
}
```

Output:
val is 66
val is 5

In a nutshell, call by reference should only be used when:

More than one value needs to be returned from a function (the return of one value is handled by a function normally, anything else must be passed back via the parameter list (call by reference).

If the value being passed is a large data structure (array), then call by reference is more efficient than call by value. This is because call by value would copy the entire structure to another location.