

ObjectDemo

```
#include <iostream>
#include <string>
#include "Thing.h"
using namespace std;

//=====
void localVar() {
    Thing t;
}
//-----
void localWithIf() {
    Thing a;
    if (false) {
        Thing b;
    } else {
        Thing c;
    }
}
//-----
void localWithLoop() {
    Thing s;

    int i=0;
    while (i<3) {
        Thing t;
        cout << "inside pass " << i << " of loop" << endl;
        i++;
    }
}
//-----
void scopeDemo() {
    localVar();
    localWithIf();
    localWithLoop();
}

//=====
void constructionDemo() {
    Thing u,v; // creates two instances with default constructor
    v.setBook("Green Eggs and Ham");
    Thing w("Snow");
    Thing x(v); // copy constructor
    u = v; // assignment operator
    v.setBook("Fox in Socks");
}

//=====
void passByVal(Thing param) {
    param.setBook("Snow");
}
//-----
void passByRef(Thing& param) {
    param.setBook("Sneetches");
}
//-----
void parameterDemo() {
    Thing t;
    t.setBook("Green Eggs and Ham");
    passByVal(t);
    passByRef(t);
}
```

ObjectDemo

```
//=====
void referencePointerDemo() {
    Thing s;
    Thing t("Green Eggs and Ham");
    Thing &u = s;    // u is not a new thing, just a new identifier associated with object
    u.setBook("Hop on Pop");
    Thing *v;
    v = &t;
    (*v).setBook("Snow");
    v->setBook("Fox in Socks");
    Thing *w = &s;
    v = w;
    v = &t;
    *v = *w;
    Thing *x;
    x = new Thing();
    Thing *y = new Thing("Snow");
    delete x;
}
//=====
```