

Script started on Tue Sep 23 09:13:36 2008

bash-2.04\$ cat base.h

```
#ifndef base_H
#define base_H
class base
{
public:
    base(int v1, int v2);
    int gety();
protected:
    int x;
private:
    int y;
};
#endif
```

bash-2.04\$ cat base.cpp

```
#include "base.h"
#include <iostream>
using namespace std;
base::base(int v1, int v2)
{
    x = v1;
    y = v2;
}
int base::gety()
{
    cout << "(base returns y) ";
    return y;
}
```

bash-2.04\$ cat derived.h

```
#ifndef derived_H
#define derived_H
#include "base.h"
class derived: public base
{
public:
    derived(int v1, int v2, int v3);
    int getx();
    int getz();
private:
    char z;
};
#endif
```

bash-2.04\$ cat derived.cpp

```
#include <iostream>
#include "derived.h"
using namespace std;
derived::derived(int v1, int v2, int v3)
    : base(v1, v2) //Calling the constructor
in base
{
    z = v3;
}
int derived::getx()
{
    cout << "(derived has access to x) ";
    return x;
}
int derived::getz()
{
    cout << "(derived owns z) ";
    return z;
}
```

bash-2.04\$ cat driver.cpp

```
#include <iostream>
#include "derived.h"
using namespace std;
int main()
{
    derived a(1,2,3);
    cout << "x = " << a.getx() << endl;
    cout << "y = " << a.gety() << endl;
    cout << "z = " << a.getz() << endl;
    return 0;
}
```

bash-2.04\$ cat makefile

```
driver: driver.o base.o derived.o
    CC -o driver driver.o base.o
    derived.o
driver.o: driver.cpp
    CC -c driver.cpp
base.o: base.h base.cpp
    CC -c base.cpp
derived.o: derived.h derived.cpp
    CC -c derived.cpp
bash-2.04$ exit
```