CS153

8/28/13

**For testing and submitting HW**

./a.out < xxxxxx.txt

./a.out < xxxxxx.txt > output.txt

cat output.txt

**\*classes with pointers**

The big-3:

1. Destructor(called when object goes out of scope)
2. Copy constructor(called on declaration with initialization or pass by value)
3. Operator “=”(called on: A=B)

**Default operator =**

Const Myclass& Mycalss::operator=(const Myclass &rhs)

{

If(this!=&rhs) //alias test

{x=rhs.x}

return this;

}

**Why default would be messed up?**

class pig

{

~~~~~~~~~~~

};

class Farm

{

public:

Pig\* pen;

Farm(int i)

{

Pen=new Pig[i];

}

};

void foo()

{

Farm a(5);

}

**Q1:When foo() ends, the class is gone, but the array that produced by the class is still there crashing memory.**

**Solution**

~Farm()

{

Delete []pen;

}

**Q2:When goes to the end**

**void foo()**

**{**

**Farm a(5);**

**Farm b(a); //shallow copy**

**b.pen[i]=x;**

**}**

**2 Pointers point to 1 array, can modify array with pointer b.**