

### // Problem 1.

// Assume any necessary #includes are completed and we are using  
// the standard namespace

```
class Int {
public:
    // constructors
    Int() : val_(0) { cout << "cons "; }
    Int(int n) { val_ = n; cout << "intcons "; }
    Int(const Int &v) : val_(v.val_) { cout << "copycons "; }

    // operations
    int get_val() {
        cout << "get_val ";
        return val_;
    }
    Int operator+(const Int &v) {
        cout << "op+ ";
        return Int(val_ + v.val_);
    }
    Int & operator=(const Int &v) {
        cout << "op= ";
        if (this != &v) { val_ = v.val_; }
        return *this;
    }
    Int & operator+=(const Int &v) {
        cout << "op+= ";
        val_ += v.val_;
        return *this;
    }
private:
    int val_; // value stored in this Int
};
```

// What does it output

```
int main() {
    Int zero;
    Int one = zero + 1;
    cout << ";\n";
    Int two = zero;
    two += one + one;
    cout << ";\n";
    cout << two.get_val() + 1;
    cout << endl;
    return 0;
}
```

**// Problem 2.**

// Do you see any problem with this code?

```
class Pointer333 {
public:
    Pointer333(int x) { x_ = new int(x); }

    Pointer333(const Pointer333 &p) {
        x_ = p.x_;
    }

    Pointer333 & operator=(const Pointer &p) {
        delete(x_);
        x_ = p.x_;
        return *this;
    }

    ~Pointer() {
        delete(x_);
    }

private:
    int *x_;
};
```

**// Problem 3.**

// Is this valid code?

```
string dept = "cse";
string course = "333";

string get(int opt) {
    return opt == 1 ? dept : course;
}

int main(int argc, char ** argv) {
    get(2) = argc > 2 ? argv[2] ? argv[0];

    return 0;
}
```

**// Problem 4.**

// We have the follow class. Write a move constructor for it

```
class OurClass {
public:
    OurClass() { x_ = nullptr; y_ = "333"; }
private:
    int *x_;
    string y_;
};
```