Const Member Functions

class MyClass {
    int x;
public:
    MyClass(int x) :(x) {}

    int GetX() {
        return x;
    }
};

Does this work?

void PrintX(const MyClass &a) {
    cout << a.GetX() << endl;
}

No, even if GetX doesn’t modify the class!
class MyClass {
    int x;
public:
    MyClass(int x) : x(x) {}
    int GetX() const {
        return x;
    }
};

void PrintX(const MyClass &a) {
    cout << a.GetX() << endl;
}

“Const” allows the member function to be used by a const instance of the class while preventing the function body from modifying the class.

Does this work? Works after adding const.
Const Member Functions

class MyClass {
    int x;

public:
    MyClass(int x) : x(x) {}

    int GetX() {
        // In GetX's point of view,
        // What is the type of x? int
        // What is the type of this? MyClass *const
        return x;
    }
};
Const Member Functions

class MyClass {
    int x;

public:
    MyClass(int x) : (x) {}  

    int GetX() const 
    {  
        // In GetX's point of view,  
        // What is the type of x?  
        // What is the type of this?  
        return x;
    }
};
class MyClass {
    int x;
public:
    MyClass(int x) : x(x) {} 
    int &GetX() {
        return x;
    }
    int GetX() const {
        return x;
    }
};

void test() {
    MyClass a(3);
    a.GetX() = 5;
    const MyClass b(3);
    b.GetX() = 5;
}

Compilation Error
Const Member Functions – Overloading

class MyClass {
    int x;
public:
    MyClass(int x) : x(x) {}
    int &GetX() {
        return x;
    }
    int &GetX() const {
        return x;
    }
};

Which version is called?

void test() {
    MyClass a(3);
    a.GetX() = 5;
    const MyClass b(3);
    b.GetX() = 5;
}
Midterm Review

• Work on the problems on the handout on your own.

• Raise your hand if you’re stuck, want clarification or to check answer.

• A more complete set of problems is available on the course website: Home – Exams – Midterm Review Packet