

Midterm Practice

1. True/False.

- a) struct members default to private while class members default to public. **False**
- b) Except for default access modifiers, a class is identical to a struct. **True**
- c) An enclosing class gets automatic access to private members of a nested class inside it. **False**
- d) friend overwrites private. **True**
- e) Destructors can't be virtual. **False**

2. What is memory leak?

Memory leak appears when memory is not deallocated properly. It will cause a system to run out of memory.

3. What are the 3 numbers the code below is printing?

```
int x, y;
int *px, *py;

int f () {
int s = *px + *py;
cout << s << endl;
return s;
}

int main() {
    x = y = 2;
    px = &x; py = &y;
    x = y = f();
    cout << f() << endl;
}
```

4
8
8

4. What are the 3 numbers the code below is printing?

```
int x, y;
int *px, *py;

int f () {
static int s = *px + *py;
cout << s << endl;
return s;
}

int main() {
    x = y = 2;
    px = &x; py = &y;
    x = y = f();
    cout << f() << endl;
}
```

4
4
4

5. What are the 6 numbers the following code fragment is printing?

```
int x, y;
int *px, *py;

int f ( int a, int& b ) {
    static int s = *px + *py;
    cout << s << endl;
    x = a + s; y = b + s;
    a = x;
    b = y;
    s = x - y;
    cout << s << endl;
    return s;
}

int main() {
    x = y = 1;
    int a = 2, b = 2;
    px = &x; py = &y;
    x = f(a, b);
    b = f(a, b);
    cout << *px << endl;
    cout << *py << endl;
    return 0;
}
```

2
0
0
-2
2
4

6. What are the 6 numbers the following code fragment is printing?

```
int x, y;
int *px, *py;

int f ( int a, int b ) {
    static int s = *px + *py;
    cout << s << endl;
    x = a + s; y = b + s;
    a = x;
    b = y;
    s = x - y;
    cout << s << endl;
    return s;
}

int main() {
    x = y = 1;
    int a = 2, b = 2;
    px = &x; py = &y;
    x = f(a, b);
    b = f(a, b);
    cout << *px << endl;
    cout << *py << endl;
    return 0;
}
```

2
0
0
0
2
2

7. Write a code fragment allowing you to add 20 C objects to a std::vector .

```
class C {
    int a, b;
public:
    C (int iA, int iB) :
        a(iA), b(iB) {}
};
```

```
// Other proper solutions are also fine.
vector<C*> v;
for ( int i=0; i<20; i++ ) v.push_back ( new C(0,0) );
```

8. What is the following code fragment printing?

```
struct A {
    public:
        A() { cout << "A\n"; }
};
class B : public A {
    public:
        B() { cout << "B\n"; }
};
int main() {
    if ( true ) { B b; }
    A* a = new B;
    delete a;
    return 0;
}
```

```
A
B
A
B
```

9. What is the following code fragment printing?

```
struct A {
    public:
        A() { cout << "A\n"; }
        ~A() { cout << "~A\n"; }
};
class B : public A {
    public:
        B() { cout << "B\n"; }
        ~B() { cout << "~B\n"; }
};
int main() {
    if ( true ) { B b; }
    A* a = new B;
    delete a;
    return 0;
}
```

```
A
B
~B
~A
A
B
~A
```

10. What is the following code fragment printing?

```
struct A {
    public:
        A() { cout << "A\n"; }
        virtual ~A() { cout << "~A\n"; }
};
class B : public A {
    public:
        B() { cout << "B\n"; }
        ~B() { cout << "~B\n"; }
};
int main() {
    if ( true ) { B b; }
    A* a = new B;
    delete a;
    return 0;
}
```

A
B
~B
~A
A
B
~B
~A

11. What is the following code fragment printing?

```
class A {
    public:
        A() { cout << "A\n"; }
        virtual ~A() { cout << "~A\n"; }
};
class B : public A {
    public:
        B() { cout << "B\n"; }
        virtual ~B() { cout << "~B\n"; }
};
class C : public B {
    public:
        C() { cout << "C\n"; }
        virtual ~C() { cout << "~C\n"; }
};

int main() {
    A* a = new B;
    if ( true ) { C c; }
    delete a;
    return 0;
}
```

A
B
A
B
C
~C
~B
~A
~B
~A

12. What is the following code fragment printing?

```
class A {
public:
    A() { cout << "A\n"; }
    virtual ~A() { cout << "~A\n"; }
};
class B : public A {
public:
    B() { cout << "B\n"; }
    ~B() { cout << "~B\n"; }
};
class C : public B {
public:
    C() { cout << "C\n"; }
    ~C() { cout << "~C\n"; }
};

int main() {
    A* a = new B;
    if ( true ) { C c; }
    delete a;
    return 0;
}
```

A
B
A
B
C
~C
~B
~A
~B
~A

13. What is the following code fragment printing?

```
class A {
public:
    A() { cout << "A\n"; }
    virtual ~A() { cout << "~A\n"; }
};
class B : public A {
public:
    B() { cout << "B\n"; }
    virtual ~B() { cout << "~B\n"; }
};
class C : public B {
public:
    C() { cout << "C\n"; }
    virtual ~C() { cout << "~C\n"; }
};

int main() {
    A* a = new B;
    if ( true ) { C c; }
    B b;
    delete a;
    return 0;
}
```

A
B
A
B
C
~C
~B
~A
A
B
~B
~A
~B
~A