1. What is the code below printing?

```cpp
int x=2;
void f () {
    static int s = x;
    cout << s << endl;
}
int main() {
    cout << x << endl;
    f();
    x = 4;
    f();
    cout << x << endl;
}
```

Result:
```
2
2
4
```

2. What is the code below printing?

```cpp
struct A {
    virtual int f ( int x ) { return (x); }
};
struct B : public A {
    int f ( int x ) { return (x+1); }
};
struct C : public B {
    int f ( int x ) { return (x+2); }
};

void main() {
    C* c = new C;
    cout << c->f(10)<< endl;
    cout << c->A::f(10)<< endl;
    cout << c->B::f(10)<< endl;
    delete c;
    B* b = new C;
    cout << b->f(30)<< endl;
    cout << b->B::f(30)<< endl;
    cout << b->A::f(30)<< endl;
    delete b;
    A* a = new C;
    cout << a->f(50)<< endl;
    cout << a->A::f(50)<< endl;
    delete a;
}
```

Result:
```
12
12
11
10
32
31
30
52
50
```

3. Write a class Vec containing two floats and with overloaded operators correctly implementing vector addition and multiplication by a scalar.

```cpp
class Vec
{
    public:
        float x, y;
        Vec (float a, float b) { x=a; y=b; }
};

Vec operator+ (Vec& v1, Vec& v2) { return Vec(v1.x+v2.x,v1.y+v2.y); }
Vec operator* (Vec& v, float d) { return Vec(v.x*d,v.y*d); }
```