Given a Java-like program as follows.

```
class A
        {
                public
                int n;
                public int f1()
                {
                        n+=4; int m= f2();return n + m + 1;
                }
                public int f2()
                {
                        n += 1;
                        return n;
                }
                public int f3()
                {
                        f1();
                        return n;
                }
        };

        class B : A
        {
                public int f1()
                {
                        n-=4; int m= f2(); return n - m;
                }
                public int f2()
                {
                        n += n;
                        return n;
                }

        };
}

…
A b = new B(); b.n = 4;
cout << b.f3(); //1
```

State the value displayed after the statement commented as //**1** executes in the following cases:

Case	static binding	dynamic binding
a	f1,f2	
b		f1,f2
c	f1	f2
d	f2	f1

SOLUTION:

a. Static binding for both f1,f2 → f1,f2 will be used from class A, not B (seen by outer scope)

```
public int f1()
{
        n+=4; int m= f2();return n + m + 1;
}
public int f2()
{
        n += 1;         //2
        return n;
}
public int f3()
{
        f1();           //1
        return n;
}
```

//1 → n += 4 → n = 8
//2 → n += 1 → n = 9

b. Dynamic binding for both f1, f2 → f1,f2 will be used from class B, not A (inter scope)

```
public int f1()
{
        n-=4; int m= f2(); return n - m;
}
public int f2()
{
        n += n;         //2
        return n;
}
public int f3()
{
        f1();           //1
        return n;
}
```

//1 → n -=4 → n = 0
//2 → n +=n → n = 0

c. Static binding for f1, dynamic binding for f2

```
public int f1()
{
        n+=4; int m= f2();return n + m + 1;
}
public int f2()
{
        n += n;         //2
        return n;
}
public int f3()
{
        f1();           //1
        return n;
}
```

//1 → n +=4 → n = 8
//2 → n +=n → n = 16

d. Static binding for f2, dynamic binding for f1

```
public int f1()
```

```
{
        n-=4; int m= f2(); return n - m;
}
public int f2()
{
        n += 1;         //2
        return n;
}
public int f3()
{
        f1();           //1
        return n;
}
```

//1 □ n -=4 □ n = 0

//2 □ n +=1 □ n = 1

But, f2 is static binding → n doesn't change when get out of scope f2 → n = 0