

```
public class A {  
    public A() {  
        System.out.println("1");  
    }  
  
    public void a() {  
        System.out.println("2");  
    }  
}
```

## 문제

(1) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)

```
public class B extends A {  
    // empty class!  
}
```

```
B b = new B();  
b.a();
```

정답:

1  
2

자식 클래스의 인스턴스 생성 시 부모 클래스의 생성자는 반드시 불리게 됩니다. 자식 클래스에서 따로 명시하지 않은 경우 부모 클래스(A)의 기본 생성자(A())가 불리기 때문에 먼저 "1" 이 출력됩니다.

그 후 b.a() 를 호출하는데요. B 는 A 의 메소드를 상속받기 때문에 A 의 a 메소드가 호출되어 "2" 가 출력됩니다.

(2) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)

```
public class C extends A {  
    public void a() {  
        System.out.println("3");  
    }  
}
```

```
C c = new C();
c.a();
```

정답:

1  
3

(1)번 문제와 같은 이유로 "1" 이 먼저 출력 됩니다. 뒤이어 `c.a()` 의 결과를 봐야 하는데, `c` 클래스에서 `A` 클래스의 `a` 메소드를 오버라이드 했네요. 그렇기 때문에 `A` 의 `a` 메소드가 호출되지 않고, `c` 의 `a` 메소드가 호출됩니다. 즉, "3" 이 출력되는 거죠.

**(3) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)**

```
public class D extends C {
    public void a() {
        System.out.println("4");
        super.a();
    }
}
```

```
D d = new D();
d.a();
```

정답:

1  
4  
3

역시 (1)번 문제와 같은 이유로 "1" 이 먼저 출력됩니다. 부모 클래스의 생성자가 불릴 때, 부모의 부모 생성자까지 모두 호출되기 때문에 `A` 의 생성자까지 호출되는 것이죠.

그리고 `D` 가 부모 클래스인 `C` 의 `a` 메소드를 오버라이드 했기 때문에 `D` 의 `a` 메소드가 호출됩니다. 따라서 일단 "4" 가 출력되고 `super.a()` 가 불립니다. `super.a()` 는 부모 클래스인 `C` 의 `a` 메소드이기 때문에 "3" 이 출력됩니다.

**(4) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)**

```
public class E extends A {

    public E() {
        System.out.println("5");
    }

    public E(int a) {
        System.out.println("6");
    }
}
```

}

```
E e = new E(1);
e.a();
```

정답:

```
1
6
2
```

`E e = new E(1)` 를 보니 파라미터가 있는 생성자를 호출하네요. 즉, `E(int a)` 를 호출하는 것이죠. 이 경우에도 부모 클래스의 생성자가 먼저 호출됩니다. 따라서 일단 "1" 이 출력되고 나서 "6" 이 출력됩니다.

그 후 `e.a()` 를 호출하는데요. `E` 는 `A` 의 메소드를 상속받기 때문에 `A` 의 `a` 메소드가 호출되어 "2" 가 출력됩니다.

(5) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)

```
public class F {
    public F(int a) {
        System.out.println("7");
    }

    public void a() {
        System.out.println("8");
    }
}
```

```
public class G extends F {
    public void a() {
        super.a();
        System.out.println("9");
    }
}
```

```
G g = new G();
g.a();
```

정답: 오류 발생

부모인 `F` 클래스에 기본 생성자가 없으면 자식 클래스에서는 어떤 생성자를 호출할지 알 수 없습니다. `super(1)` 과 같이 명확하게 호출해주어야 합니다. 그렇기 때문에 문법 오류가 발생합니다. IDE에 직접 써보시면 오류가 나오는 것을 확인하실 수 있습니다.

(6) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)

```
public class H extends E {
    public H(int a) {
        System.out.println("10");
    }
}
```

```
H h = new H(1);
h.a();
```

**정답:**

```
1
5
10
2
```

**A** -> **E** -> **H** 순으로 상속이 이루어졌습니다. 가장 먼저 **A**의 생성자가 호출되어 "1"이 출력되겠네요.

그 후에 **E**의 생성자가 호출되어야 하는데, 따로 표시하지 않았기 때문에 **E**의 기본 생성자가 호출됩니다. 즉, "5"가 출력됩니다.

마지막으로 **H**의 생성자가 호출되어 "10"이 출력되죠.

상속이 이루어지는 가운데 **a** 메소드에 대한 오버라이드가 따로 없었기 때문에 **A**의 **a**가 호출되어 "2"가 출력됩니다.

**(7) 출력되는 값은 무엇입니까? (오류가 나올 수도 있습니다.)**

```
public class I extends A {
    public I() {
        System.out.println("11");
        super();
    }
}
```


```
I i = new I();
i.a();
```


**정답: 오류 발생**

부모 클래스의 생성자 호출은 자식 클래스 생성자의 맨 첫 줄에서 불러야 합니다. 따라서 오류가 발생합니다.



수업을 완료하셨으면 체크해주세요.

 수강생 Q&A 보기

 [/questions?](/questions?assignment_id=489&sort_by=popular) 질문하기

[assignment\\_id=489&sort\\_by=popular](/questions/new?assignment_id=489&sort_by=popular)

[assignment\\_id=489&op1=%EA%B0%9D%EC%B2%B4+%EC%A7](#)

< [이전 강의](/assignments/488)  
super 퀴즈 (/assignments/488)

[다음 강의](/assignments/421)  
protected 접근 제어자 > (/assignments/421)