



Question 1

Q. Inheritance 연습문제

```
package Inheritance;

class Superclass {

    // fields
    int a = staticCall("1번입니다.");
    static int b = staticCall("2번입니다.");

    // static method
    static int staticCall(String msg) {
        System.out.println(msg);
        return 100;
    }

    // constructor
    public Superclass() {
        staticCall("3번입니다.");
    }

    public Superclass(int i) {
        this();
        staticCall("4번입니다.");
    }

    // method

    public void myFunc() {
        System.out.println("5번입니다.");
    }
}

public class InheritanceTest extends Superclass {
    // fields
    int c = staticCall("6번입니다.");
    static int d = staticCall("7번입니다.");

    // constructor
    public InheritanceTest() {
        super(100);
        staticCall("8번입니다.");
        super.myFunc();
    }

    @Override
```

```

public void myFunc() {
    System.out.println("9번입니다.");
}

public static void main(String[] args) {
    System.out.println("10번입니다.");
    Superclass obj = new InheritanceTest();
    obj.myFunc();
}
}

```

A. 정답.

```

2번입니다.
7번입니다.
10번입니다.
1번입니다.
3번입니다.
4번입니다.
6번입니다.
8번입니다.
5번입니다.
9번입니다.

```

풀이

Inheritance 클래스를 올릴려면 자신의 상위 클래스인 super클래스 부터 메소드 영역에 올라간다

메소드 영역에 static b의 공간이 만들어진다. 바로 스텍 메소드가 호출이 되면서 2번입니다가 출력

나머지는 아직 수행이 될 수가 없다. 메소드 영역에 클래스에대한 정보만 올라가는 단계이므로,

이제 Inheritance에 대한 정보를 올린다. 그래서 7번입니다가 출력된다.

main메소드가 호출 그래서 10번입니다가 출력

superclass obj = new InheritanceTest(); 인스턴스 만들어서 생성자 호출하러 간다

public InheritanceTest()로 간다

super(100) = 상위 클래스의 컨스트럭터를 호출 숫자를 받는놈 인자를 안받는놈

숫자를 받는 컨스트럭터를 호출 this(); → 내가 가지고 있는 다른 컨스트럭터를 호출하래

즉 인자를 안받는 컨스트럭터 호출 즉 3번이 찍힐것같은데? 생성자가 만들어지기 전에 필드에 대한 공간이 먼저 만들어져야 초기화가 가능하기때문에 `int a = staticcall` (1번입니다)가 출력된다.

그다음 3번 4번 여기까지는 오케이 instance영역 호출이 되어서 c라는 공간이 만들어지고 6번 호출

그다음 8번 호출 그다음 `super.myfunc`기 때문에 상위 클래스에 `myfunc` 5번이 출력

상위 클래스 타입 obj???? 이건 뭐지??

다시한번 5번이 찍힐것같은데 객체에 대한 타입이 상위 타입이라 할지라도 만약 오버라이딩된 메서드가 하위에 존재한다면 메서드는 무조건 오버라이딩된 메소드를 사용한다.

→ 이것을 동적 바인딩이라고 얘기한다(Daynamic binding)

하지만 필드에 대해서는 그렇지 않다. 메서드에 대해서만 적용