

# Examples

WEEK 12

# What is the error???

```
class A {  
    public A(int x) {  
    }  
}  
class B extends A {  
    public B() {  
    }  
}  
public class Test {  
    public static void main(String[] args) {  
        B b = new B();  
    }  
}
```

Since you are invoking no-arg constructors, you will need a no-arg constructor of class A, too.

# What is the output???

```
public class Test {  
    public static void main(String[] args) {  
        A a = new A(3);  
    }  
}  
class A extends B {  
    public A(int t) {  
        System.out.println("A's constructor is invoked");  
    }  
}  
class B {  
    public B() {  
        System.out.println("B's constructor is invoked");  
    }  
}
```

Output: (It first calls the implicitly defined constructor of B: super())  
B's constructor is invoked  
A's constructor is invoked

```
public class Test{  
    public static void main(String[] args){  
        B b = new B(3,2);  
        System.out.println("Area=" + B.getArea());  
    }  
}
```

Should be **b.getArea()**  
since non-static

```
class Circle {  
    private double radius;  
  
    public Circle(double radius) {  
        radius = radius;  
    }  
  
    public double getRadius() {  
        return radius;  
    }  
  
    public double getArea() {  
        return radius * radius * Math.PI;  
    }  
}
```

**this.radius = radius;**

```
class B extends Circle {  
    private double length;
```

```
    B(double radius, double length) {  
        Circle(radius);  
        length = length;  
    }  
    this.length = length;
```

**super(radius);**  
Because while calling the  
superclass constructor you  
can't use the class name

```
@Override  
    public double getArea() {  
        return getArea() * length;  
    }  
}
```

**return super.getArea() \* length;**  
Because overridden.

**Find 5 errors!**