

When are Java super() constructors called?

Question

Consider two classes A and B:

```
public class A{
    public A(){
        // lots of constructor activity.
    }
}
```

```
public class B extends A{
    public B(){}
}
```

There are no other constructors for either A or B.

Does the constructor for A get called when a B is constructed?
Or do I need to add the following line to get A constructed?

```
public B{
    super();
}
```

Answer

There are two bizarre Java rules:

1. If a java class has no explicit constructor it is assumed to have an "implicit" empty public no-arg constructor:

```
public MyClass(){} 
```

2. If a constructor does not explicitly call either the super constructor or another of its own constructors (using `this(args)`) it is assumed to have an "implicit" call to `super()` as the first line of the constructor. Note also that `this()` or `super()` can only be called as the first line of a constructor.

Relying on either of these is generally considered to be in bad taste, so you should always have at least one constructor, and you should always explicitly call the super class's constructor.