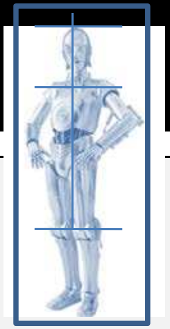
Constructor Chains

- taking a closer look, we note that **all super** class constructors of a subtype **must be called** prior to the execution of the subtype's constructor (since it may rely upon fields initialised in the super classes)
- if the programmer does not select a **super** call as the first thing in a constructor, Java makes sure the chain is **automatically triggered** if needed with the default constructors of no argument and an empty body (you will need to provide one if you have defined your own constructors)
- the constructor calls **form a chain** upwards in the type hierarchy
- once each constructor has finished initialisation, the subtype begins its initialisation



```
class Robot {
  ...
  Robot(String name) {
    this.name = name;
    numLegs = 2;
    powerLevel = 2.0f;
  }

  Robot() {
    this("Standard Model");
  }
  ...
}
```

attribute is initialised in super class Robot

```
abstract class AbstractRobot extends Robot {
  ...
}
```

```
class TranslationRobot extends AbstractRobot {
  ...
}
```

```
public class InheritanceWorld {

  public static void main (String[] args) {
    Robot c3po = new Robot();
    TranslationRobot c4po =
      new TranslationRobot("e");
    ...
    c3po.charge(10);
    c4po.charge(10);
    ...
} }
```

instantiation triggers constructor chain calling also the Robot constructor

name was initialised by super class

```
$
Standard Model charges
Standard Model charges double
```