

```
public class Soap {
  ...
}
```

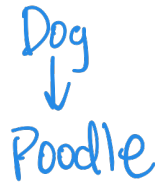
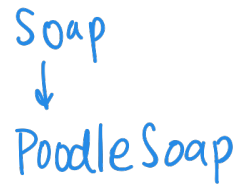
```
public class PoodleSoap extends Soap {
  ...
}
```

```
public class Dog {
  ...
}
```

```
* public void wash (Soap s) {
  System.out.println ("Dog is clean!");
}
```

```
public class Poodle extends Dog {
  ...
}
```

```
* public void wash (PoodleSoap s) {
  System.out.println ("Poodle is oodles clean!");
}
```



allowed to pass in PoodleSoap

overriding: method signature is same!
 same name
 same parameters

overloading!

```
P Poodle fifi = new Poodle ();
  fifi.wash (new Soap()); Dog is clean!
  fifi.wash (new PoodleSoap()); Poodle is oodles clean!
```

```
Dog froufrou = new Poodle ();
  froufrou.wash (new Soap()); Dog is clean!
```

```
froufrou.wash (new PoodleSoap()); Dog is clean!
```

	Static	Dynamic
fifi	Poodle	Poodle
froufrou	Dog	Poodle

Compiler Notes

1. wash (Soap)
2. wash (PoodleSoap)
3. wash (Soap)
4. wash (Soap) *

Summary

- Compiler is a cautious proofreader
- Compiler only knows static type

Method Procedure

1. During compile time, look in static type for method
2. Write down exact method signature including name and parameters
3. During run-time, only override if method signature is the same including parameters.