late binding

:-)

Late Binding

```
public class L {
    // L(): default constructor
    public L() {
    }

    // f(): facilitator
    public void f() {
        System.out.println("Using L's f()");
        g();
    }

    // g(): facilitator
    public void g() {
        System.out.println("using L's g()");
    }
}
```

```java
public class M extends L {
    // M(): default constructor
    public M() {
        // no body needed
    }

    // g(): facilitator
    public void g() {
        System.out.println("Using M's g()");
    }

    // main(): application entry point
    public static void main(String[] args) {
        L l = new L();
        M m = new M();
        l.f();
        m.f();
        return;
    }
}
```

```java
public class M extends L {
    // M(): default constructor
    public M() {
        // no body needed
    }

    // g(): facilitator
    public void g() {
        System.out.println("Using M's g()");
    }

    // main(): application entry point
    public static void main(String[] args) {
        L l = new L();
        M m = new M();
        l.f();
        m.f();
        return;
    }
}
```

Output

```
Using L's f()
using L's g()
Using L's f()
Using M's g()
```