

INF 212

ANALYSIS OF PROG. LANGS

INVERSION OF CONTROL

PUBLISH-SUBSCRIBE

Instructors: Crista Lopes

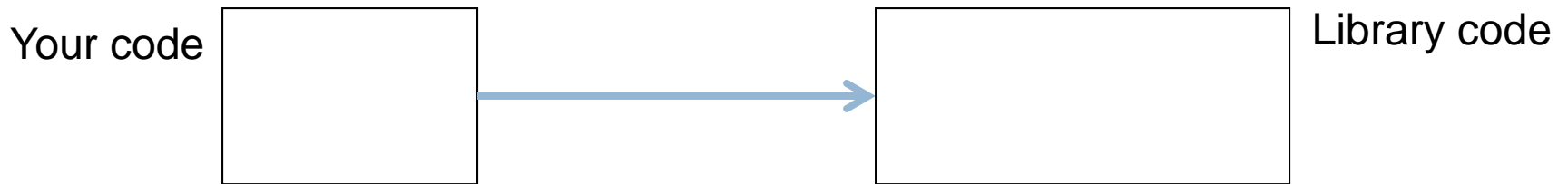
Copyright © Instructors.



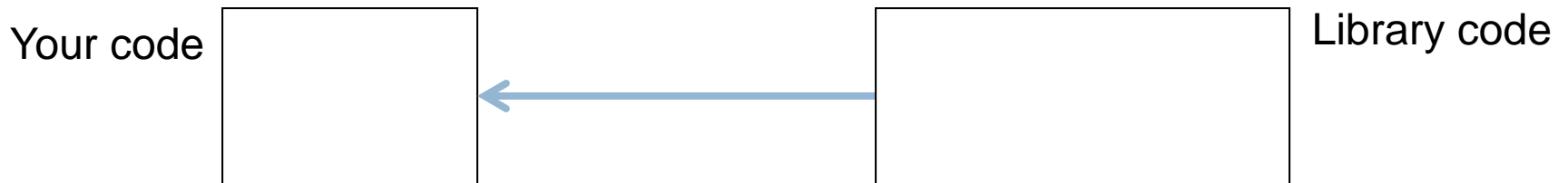
Inversion of Control

Inversion of Control (IoC)

- Usually your code calls the library code



- IoC: Library code calls your code



IoC

- Accumulation of callbacks to be called at specific times
- Popularized in OOP frameworks in the 80s
- Beware of posts on the Internet...
 - ▣ *Such words! Much confusion! Wow!*
 - ▣ Most describe dependency injection


IoC enablers

- In general:
 - ▣ Higher-order functions
 - ▣ Objects
- In statically-typed languages:
 - ▣ Abstract Data Types
 - ▣ Inheritance

Callbacks

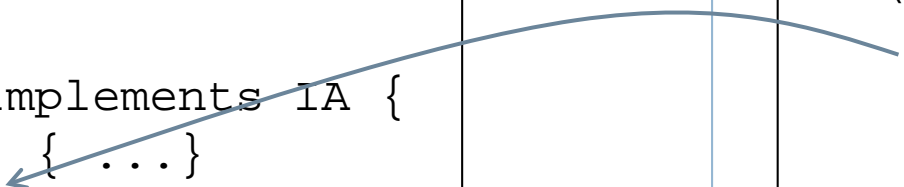
```
main() {  
    libraryFunction(myFunction);  
}  
  
myFunction(a, b) {  
    ...  
}
```

```
libraryFunction(func) {  
    ...  
    func(1, 2);  
    ...  
}
```

A blue arrow originates from the line `func(1, 2);` in the `libraryFunction` box and points to the line `myFunction(a, b)` in the `main` box, illustrating a callback call.

```
main() {  
    IA myObj = new A();  
    lib.f(myObj);  
}  
  
class A implements IA {  
    m(a, b) { ... }  
}
```

```
interface IA {...}  
  
class Lib {  
    f(IA a) {  
        ...  
        a.m(1, 2);  
        ...  
    }  
}
```

A blue arrow originates from the line `a.m(1, 2);` inside the `f` method of the `Lib` class in the right box and points to the `m(a, b)` method definition in the `A` class in the left box, illustrating a callback call.

What is a framework?



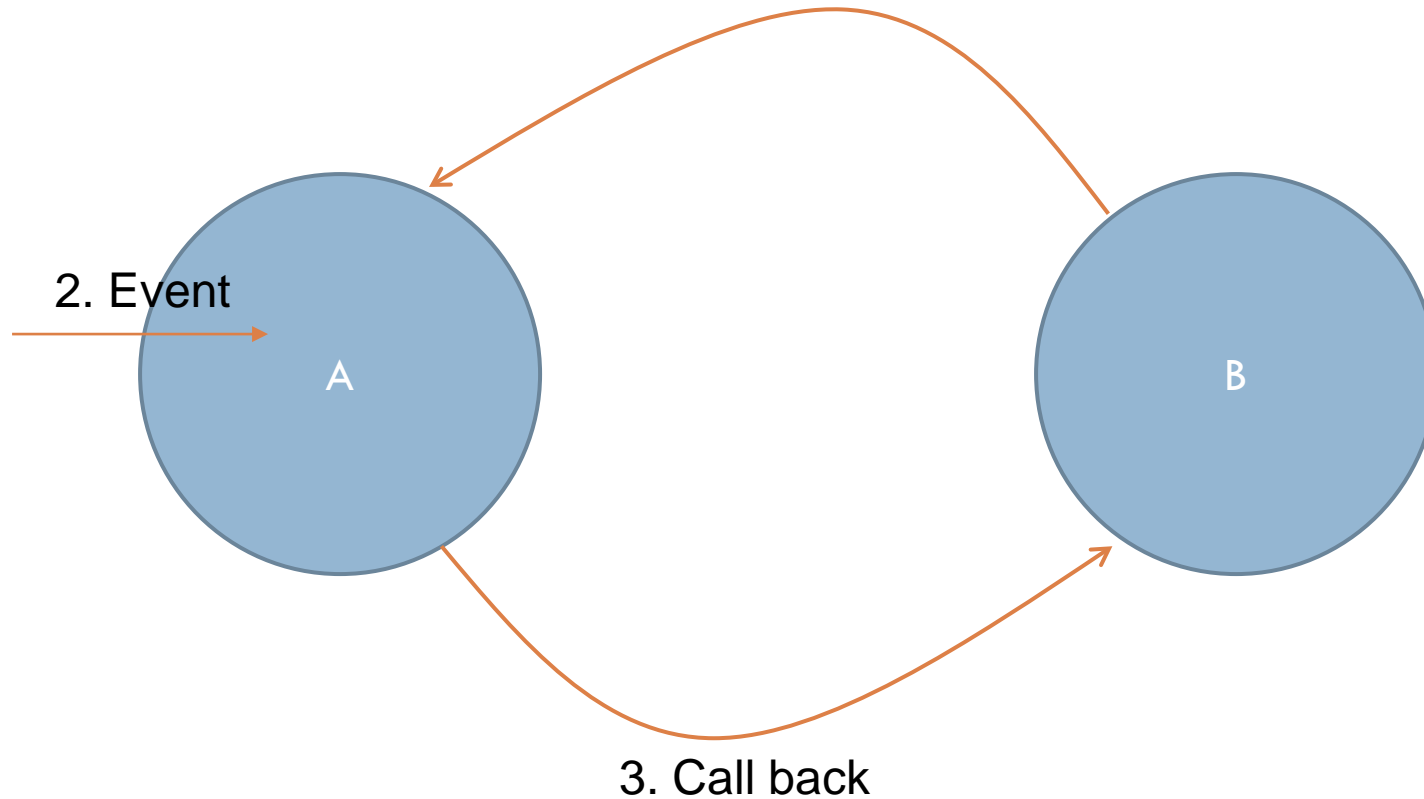
- More than a library, a generic component or even application
- Parts of it are customized by user's code via inversion of control
- Allows developers to build complete components/applications, with small customizations on deployment sites



Event-Driven

Basic Idea

1. Register for event, i.e. “call me back when it happens”



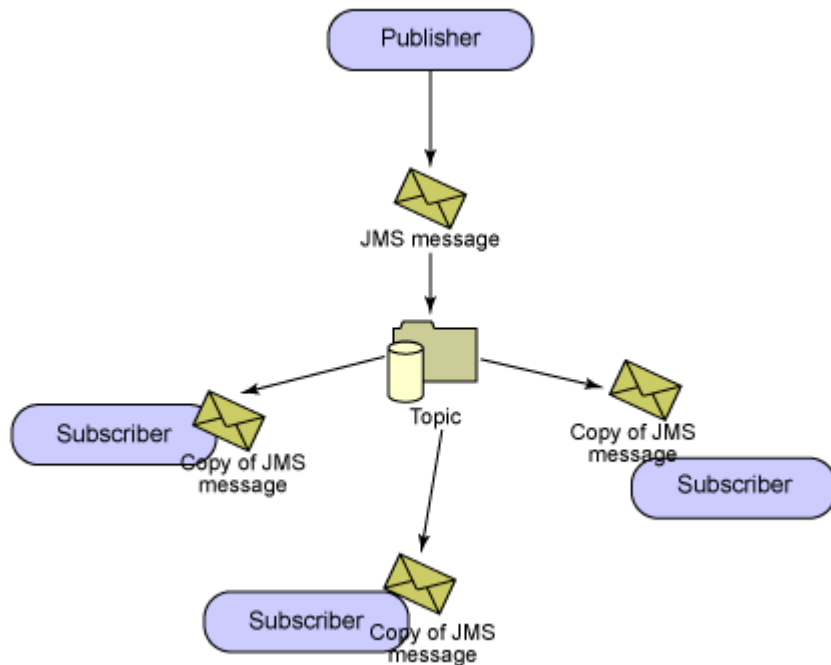
Event-driven frameworks

- Popularized by GUIs
- Popularized by OOP
- Typically OO, but OO not necessary
 - ▣ In FP, higher-order functions are the handlers

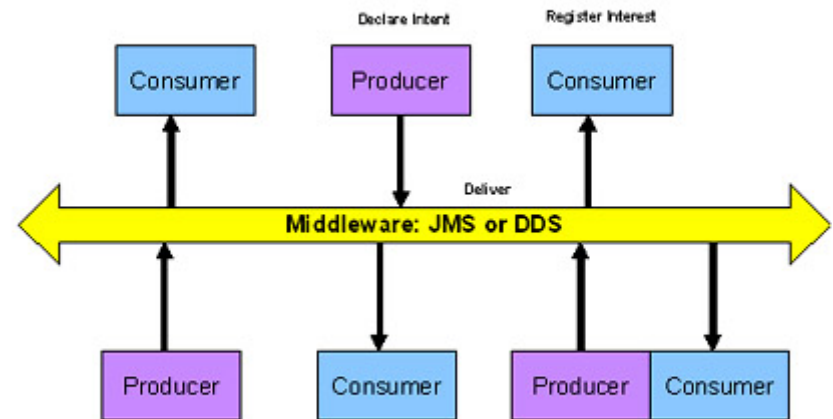


Publish-Subscribe

Publish Subscribe



Publish-Subscribe Middleware



Decouples Producers and Consumers

Pub Sub

- Producers
 - ▣ Publish events
- Consumers
 - ▣ Register for events
- Decouples components