Frameworks

15-413: Introduction to Software Engineering
Jonathan Aldrich
With thanks to George Fairbanks

Hello, World as Java Applet

```java
import java.awt.Graphics;
import java.applet.Applet;
import java.awt.Graphics;
public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
    public Applet() {
        super("Hello World!", 50, 25);
    }
}
```

Hello, World in C

```c
#include <stdio.h>
main() {
    printf("Hello World\n");
}
```

What context does the programmer need to know?
- When the programmer chooses to call printf() it will behave according to its specification

Frameworks

- Definition: “The skeleton of an application that can be customized by an application developer” – Ralph Johnson
- Characteristics
  - OO world: subclass parts of the framework
  - Specialized to a class of software
  - Hollywood model: “Don’t call us…”

Frameworks are Everywhere

- Java
  - Applets, Servlets, EJBs, Eclipse, NetBeans
- C/C++
  - Microsoft Foundation Classes, KDE, Gnome, .NET (?)
- Objective-C
  - Cocoa, Carbon (?)

Framework Benefits

- Code reuse
  - Probably even more valuable than libraries
- High productivity
  - A few lines go a long way
  - Once you know the framework!
- Standardization
  - Platform look and feel
  - Integration between components
Framework Challenges

- How to do X?
  - Must subclass C, override method m...
  - May need a sequence of method calls to get to the right object
  - e.g. Swing threading goes both ways
- Distribution
  - Must coordinate several extension points
    - "e.g. to add a context menu in Eclipse, you have to create a MenuManager in your view and a MenuListener that populates the menu when the user right-clicks.”
    - It’s not obvious that you need the MenuListener (you could write non-threading code without it), but if you forget it then things might not work
- Constraints
  - Must invoke super() when overriding
  - Can’t call method X/Y from here
  - E.g.: "Swing threading goes both ways"
  - Must handle certain events
    - App mysteriously fails if you forget
  - Make calls in sequence
    - "E.g. createPartControl in Eclipse for View setup"

Coping with Frameworks

- Read books
  - Good for an introduction
  - Good for reference
  - Too tedious to read beginning to end
- Read/copy source code
  - In practice, this is what people do most often
  - Can be hard to find code that is relevant; use search engines and book indexes
  - Danger that the code you find may be inappropriate for your problem
- Wizards
  - Magic code that works the first time
  - Limited to special circumstances
  - Still must learn the library if you want to modify it
- Read the framework code
  - Usually hopeless except for very simple cases

Frameworks vs. Patterns

- Frameworks are more concrete
  - Actual code for a particular domain
  - Design patterns aren’t code; they are a pattern that can be applied in code
- Frameworks are larger
  - May contain multiple patterns
- Frameworks are specialized to a domain
  - Design patterns apply in any domain