

# Client-Side Options (Part 1)

Copyright © 2025 by  
Robert M. Dondero, Ph.D.  
Princeton University

# Objectives

- We will cover:
  - Desktop app programming
  - A Penny **desktop** client

# Objectives

- More specifically...
- We will cover:
  - “High-level” desktop app programming
- We will not cover:
  - “Low-level” desktop app programming

# Motivation

- **Question:** Why study desktop app programming?
- **Answers:**
  - Mainstream “advanced programming”
  - Illustrates event-driven programming
  - Illustrates concurrent programming
  - Important!

# Agenda

- **Aside: Multithreaded programming**
- Java Swing
- Baseline
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

# Multithreaded Programming

- Recall **spawning.py**

```
$ python spawning.py
blue
blue
blue
blue
blue
blue thread terminated
red
red
red
red
red
red thread terminated
main thread terminated
$
```

```
$ python spawning.py
blue
blue
blue
blue
red
red
red
red
red thread terminated
blue
main thread terminated
blue thread terminated
$
```

# Multithreaded Programming

- See **Spawning1.java**

```
$ java Spawning1
main thread terminated
blue
blue
blue
blue
blue
blue
red
red
red
red
red
red thread terminated
blue thread terminated
$
```

```
$ java Spawning1
blue
blue
blue
blue
blue
main thread terminated
red
red
red
red
red
red thread terminated
blue thread terminated
$
```

# Multithreaded Programming

- **Problem:**
  - Java does not allow multiple inheritance
  - What if class `PrinterThread` already inherits from some class other than `Thread`?
- **Solution:** ...



# Multithreaded Programming

- See **Spawning2.java**

```
$ java Spawning2
blue
blue
blue
blue
blue
main thread terminated
red
red
red
red
red
blue thread terminated
red thread terminated
$
```

```
$ java Spawning2
blue
blue
main thread terminated
red
red
red
red
red
blue
blue
blue
blue thread terminated
red thread terminated
$
```

# Multithreaded Programming

- **Note:**

- `bluePrinterRunnable` is an object of a class that implements an interface that defines one method
- `bluePrinterRunnable` can be replaced with a lambda expression
- Same for `redPrinterRunnable`

- **And so...**

# Multithreaded Programming

- See **Spawning3.java**

```
$ java Spawning3
main thread terminated
blue
blue
blue
blue
blue
blue thread terminated
red
red
red
red
red
red thread terminated
$
```

```
$ java Spawning3
main thread terminated
blue
blue
blue
red
red
red
red
red thread terminated
blue
blue
blue thread terminated
$
```

# Agenda

- Aside: Multithreaded programming
- **Java Swing**
- Baseline
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

# Java Swing

Cross-platform desktop app libraries:

| Desktop App Library | Language      | Platform                            |
|---------------------|---------------|-------------------------------------|
| <b>GTK</b>          | C, C++        | Linux, Windows, macOS               |
| <b>Qt</b>           | <b>C, C++</b> | <b>Linux, Windows, macOS</b>        |
| <b>wxWidgets</b>    | C, C++        | Linux, Windows, macOS               |
| <b>Flutter</b>      | C, C++, Dart  | Linux, Windows, macOS, Android, iOS |
| <b>PyQt</b>         | Python        | Linux, Windows, macOS               |
| <b>Tkinter</b>      | Python        | Linux, Windows, macOS               |
| <b>AWT</b>          | Java          | Linux, Windows, macOS               |
| <b>Swing</b>        | Java          | Linux, Windows, macOS               |
| <b>JavaFX</b>       | Java          | Linux, Windows, macOS               |
| ...                 | ...           | ...                                 |

# Java Swing

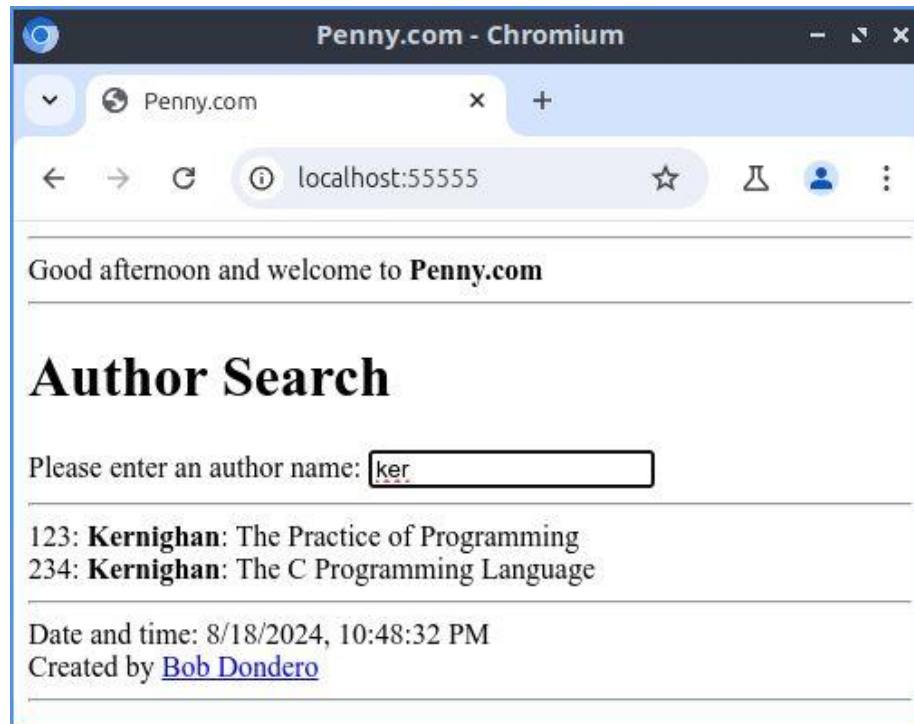
- Why study Java Swing
  - (Instead of some other GUI library)
  - I know it!
  - Popular
  - Bridges to Android programming

# Agenda

- Aside: Multithreaded programming
- Java Swing
- **Baseline**
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

# Baseline

- See **PennyJson** app
  - (Almost) same as **PennyFlaskSpa** app from previous lecture





# Baseline

- See **PennyJson** app (cont.)
  - runserver.py
  - penny.sql, penny.sqllite
  - database.py
  - index.html
    - Irrelevant for this lecture
  - **penny.py**
    - Added IO\_DELAY via an environment variable
      - Simulates slow DB
      - Shows how client performs with I/O bound server

# Agenda

- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - **Version 1: Simple**
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

# Penny Desktop v1

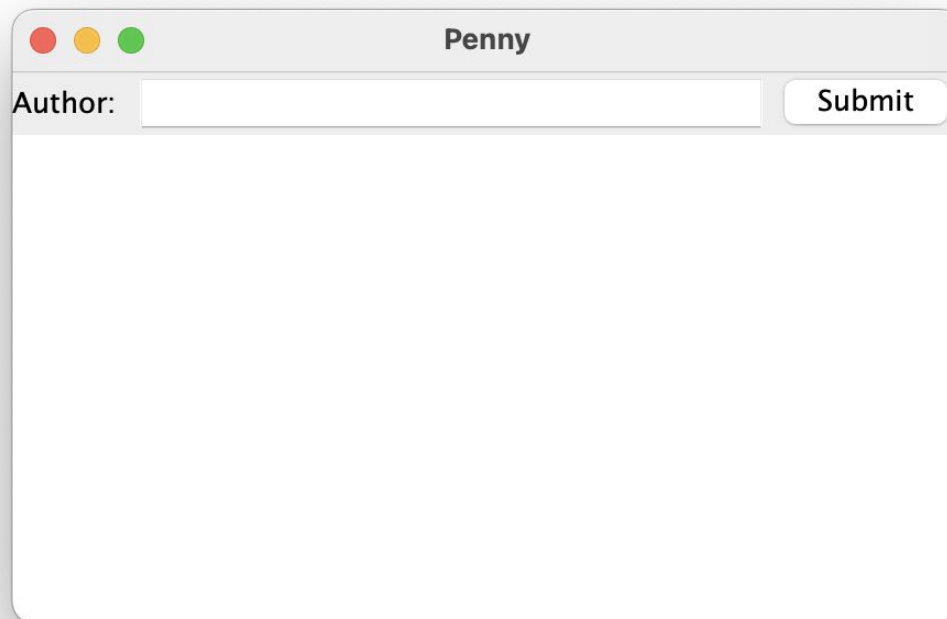
- Penny desktop v1

```
$ export IO_DELAY=1  
$ cd PennyJson  
$ python runserver.py 55555
```

```
$ cd PennySwing1  
$ ./runclient http://localhost:55555
```

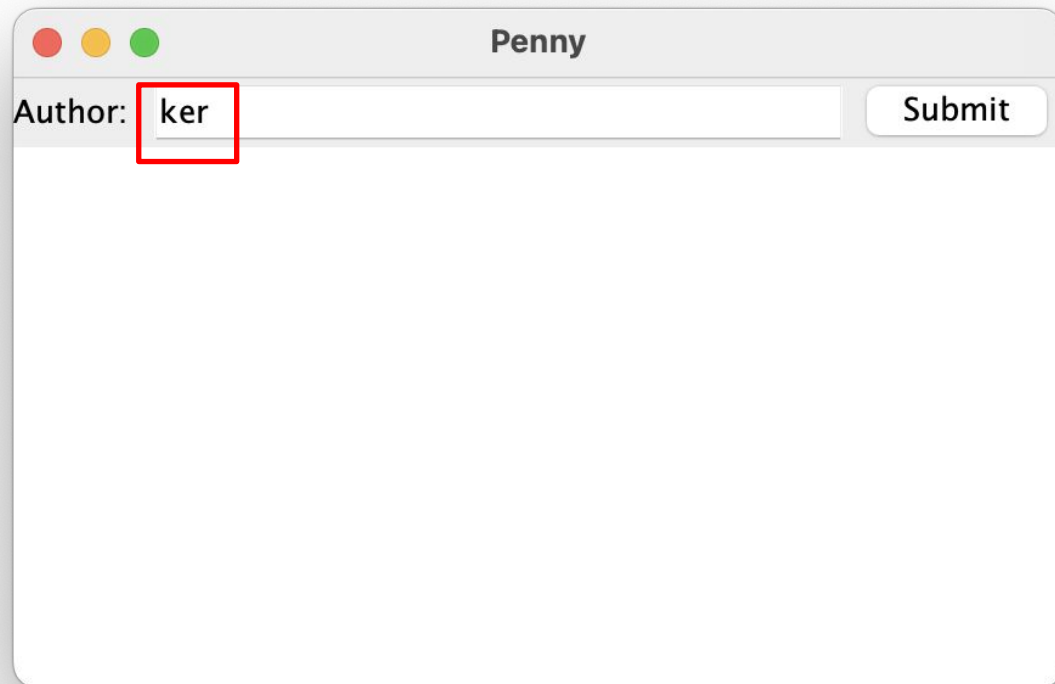
# Penny Desktop v1

- Penny desktop v1 (cont.)



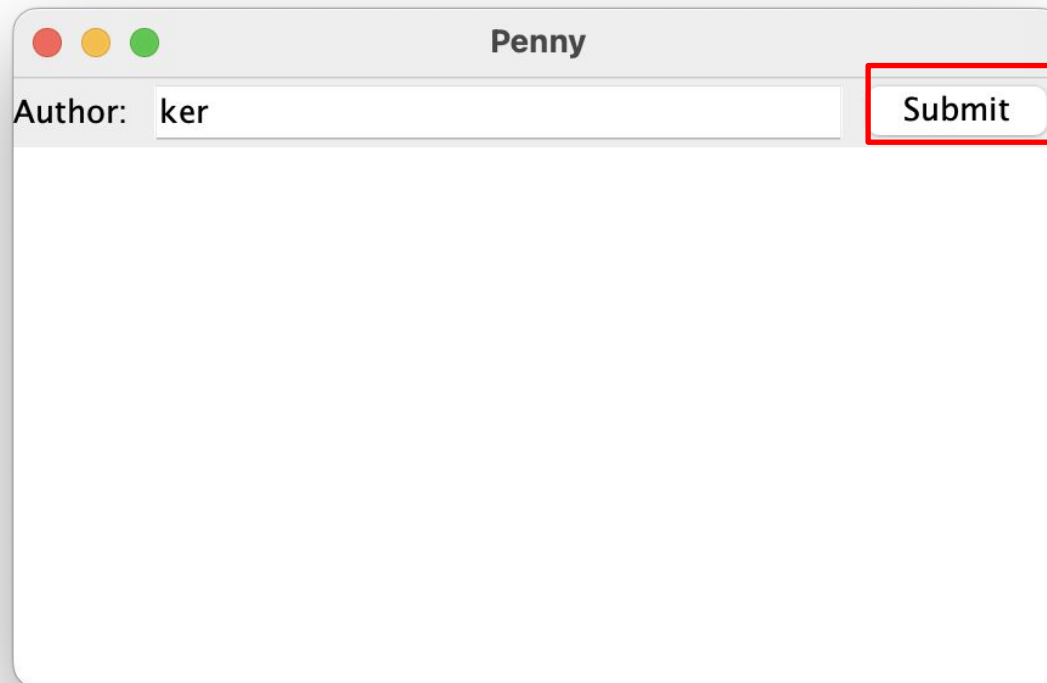
# Penny Desktop v1

- Penny desktop v1 (cont.)



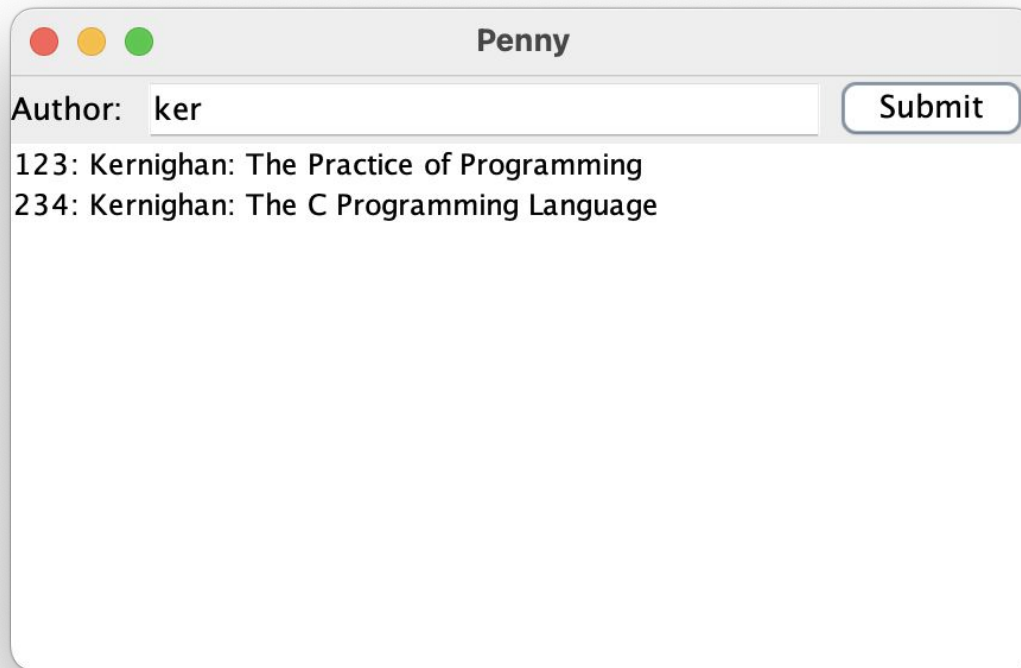
# Penny Desktop v1

- Penny desktop v1 (cont.)



# Penny Desktop v1

- Penny desktop v1 (cont.)



The screenshot shows a window titled "Penny" with a standard macOS-style title bar (red, yellow, green buttons). Inside the window, there is a text input field labeled "Author:" containing the text "ker". To the right of the input field is a "Submit" button. Below the input field, there is a list of two items:

- 123: Kernighan: The Practice of Programming
- 234: Kernighan: The C Programming Language

After a  
1 second  
delay

# Penny Desktop v1

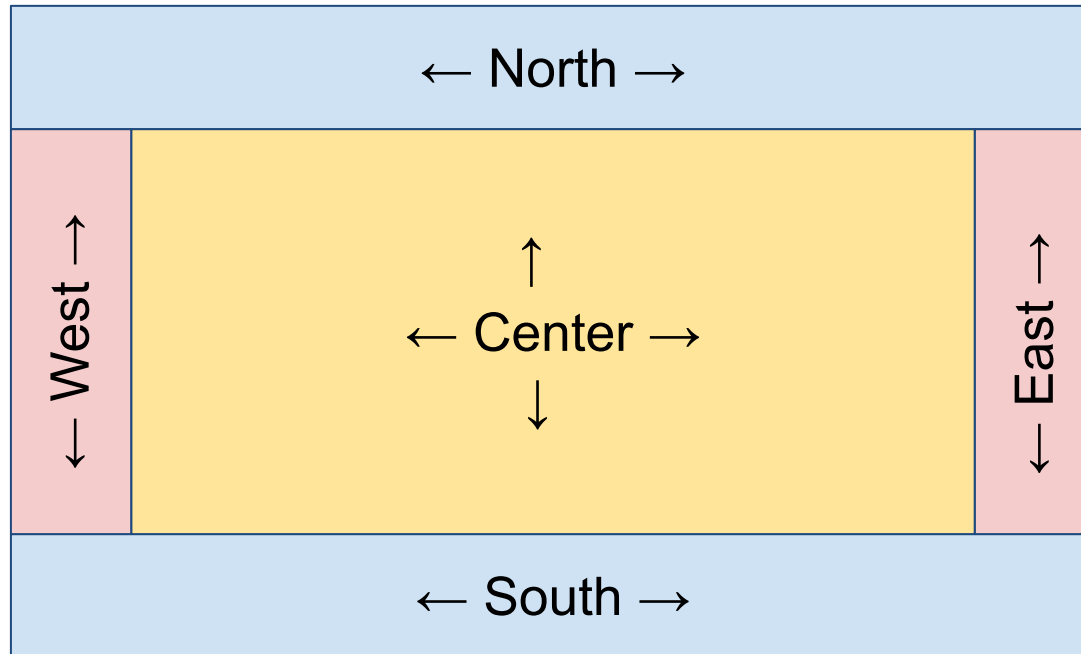
- Penny desktop v1 (cont.)
  - See **PennySwing1 app**

```
PennySwing1
  pom.xml
  runclient
  runclient.bat
  src
    main
      java
        edu
          princeton
            penny
              PennyDesktop.java
              PennyDesktopRunnable.java
```



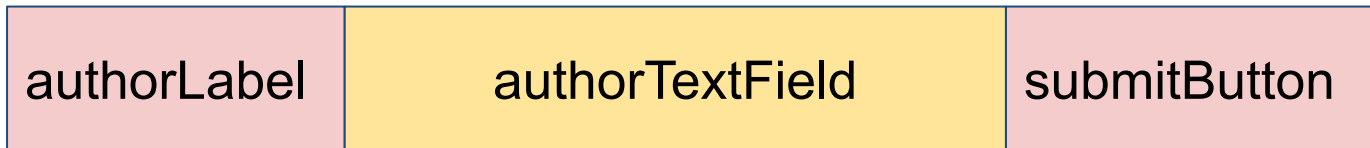
# Penny Desktop v1

The Java Swing `BorderLayout` layout manager

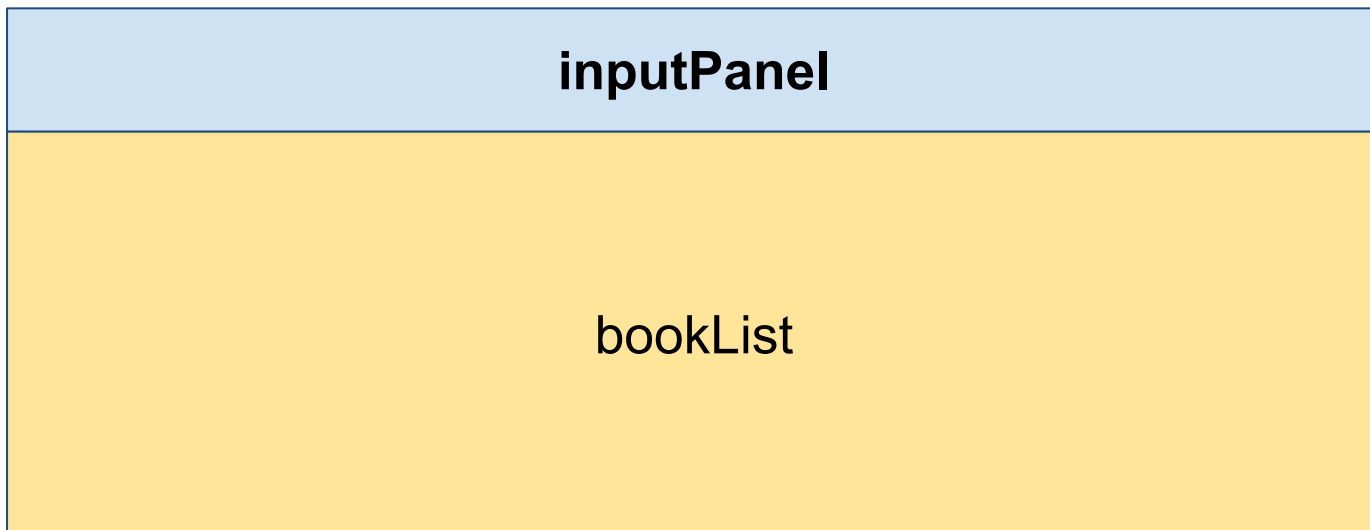


# Penny Desktop v1

## inputPanel



## frame



# Agenda

- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - **Version 2: Immediate**
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

# Penny Desktop v2

- Recall Penny desktop v1
  - **(maybe) Problem:**
    - Inconsistent window state after typing author and before clicking Submit button
  - **Solution: redesign...**
    - Eliminate Submit button
    - GUI refreshes with each keystroke

# Penny Desktop v2

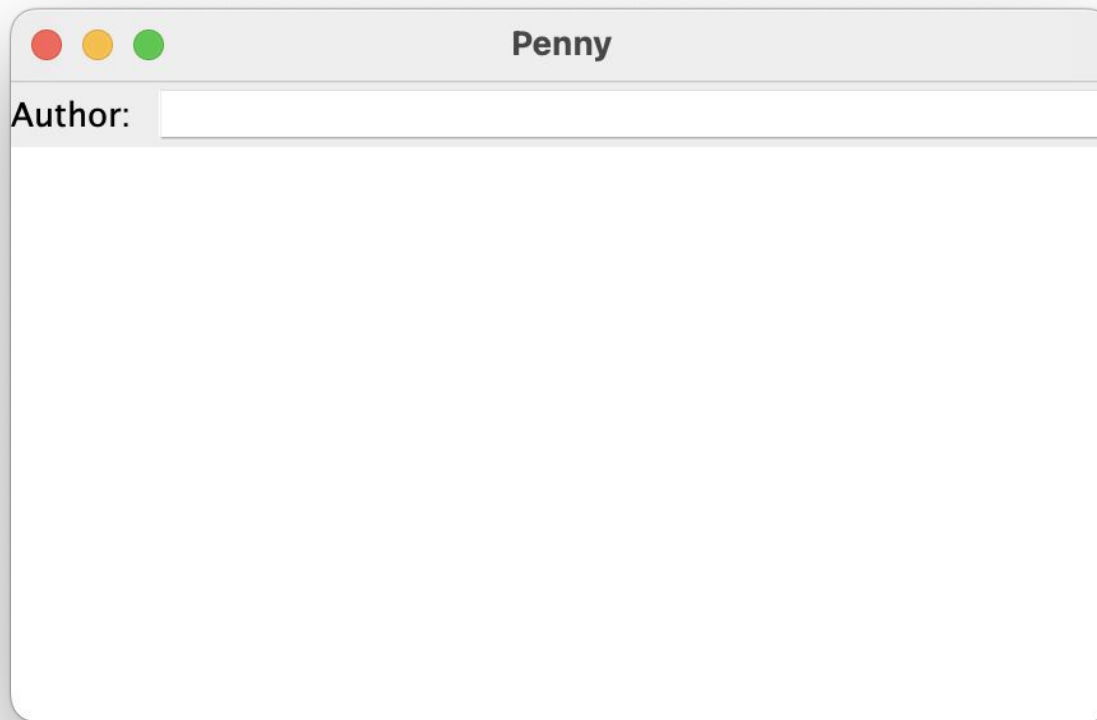
- Penny desktop v2

```
$ export IO_DELAY=1  
$ cd PennyJson  
$ python runserver.py 55555
```

```
$ cd PennySwing2  
$ ./runclient http://localhost:55555
```

# Penny Desktop v2

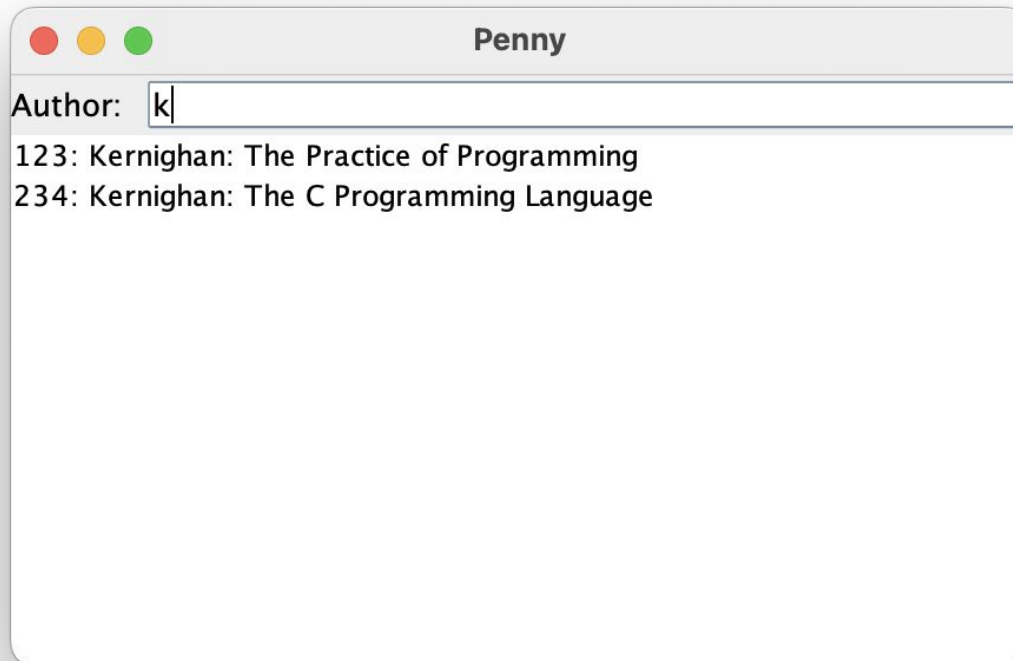
- Penny desktop v2 (cont.)



No  
Button

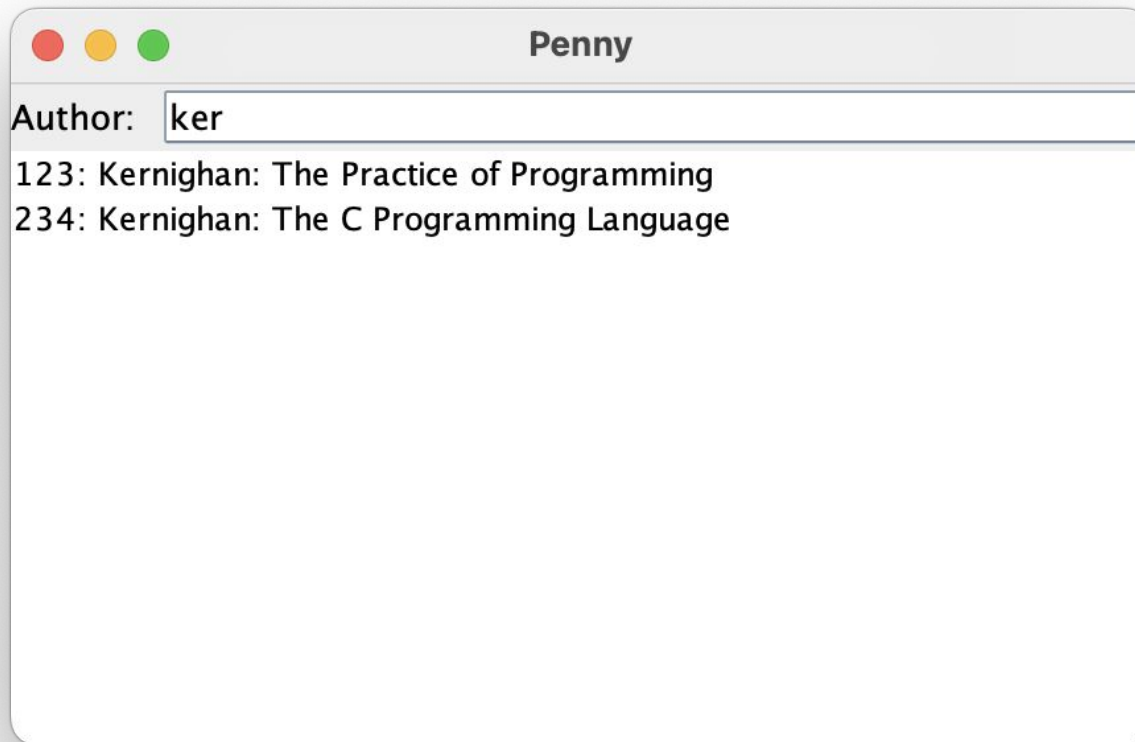
# Penny Desktop v2

- Penny desktop v2 (cont.)



# Penny Desktop v2

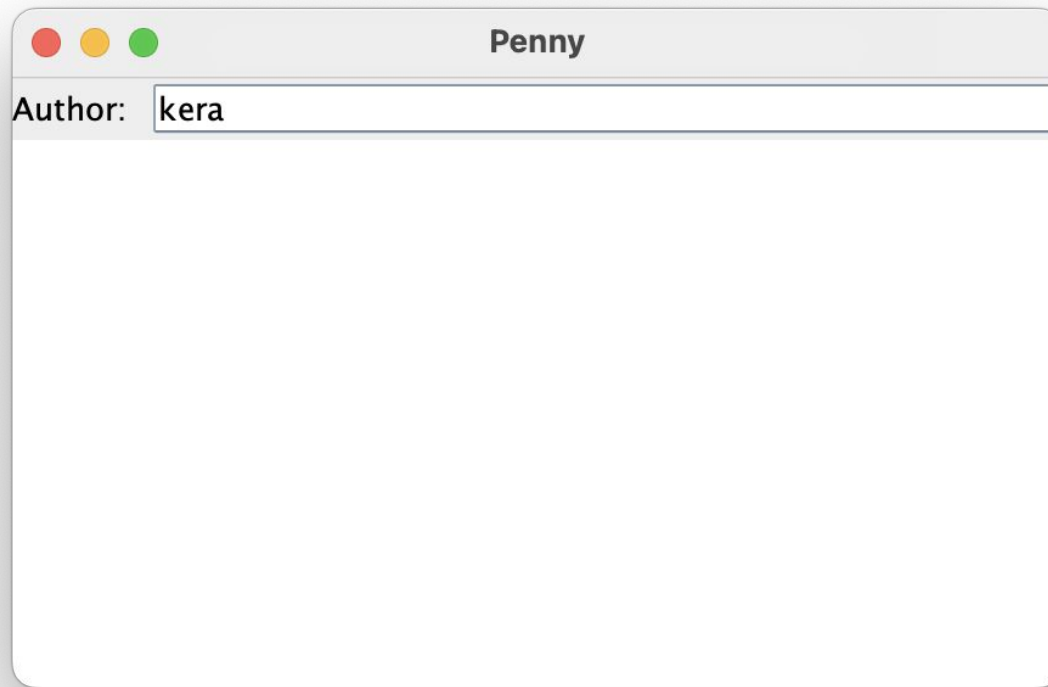
- Penny desktop v2 (cont.)





# Penny Desktop v2

- Penny desktop v2 (cont.)



# Penny Desktop v2

- Penny desktop v2 (cont.)
  - See **PennySwing2** app

```
PennySwing2
  pom.xml
  runclient
  runclient.bat
  src
    main
      java
        edu
          princeton
            penny
              PennyDesktop.java
              PennyDesktopRunnable.java
```

# Agenda

- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - **Version 3: Threads**
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

# Penny Desktop v3

- Recall Penny desktop v2
  - **Problem:**
    - Serious GUI lag
  - **Solution: redesign...**
    - Use threads!!!

# Penny Desktop v3

- Penny desktop v3
  - See **PennySwing3** app

```
PennySwing3
  pom.xml
  runclient
  runclient.bat
  src
    main
      java
        edu
          princeton
            penny
              PennyDesktop.java
              PennyDesktopRunnable.java
              WorkerThread.java
```

# Agenda

- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - **Version 4: ThreadsProper**
  - Version 5: Stop
  - Version 6: Debounce

# Penny Desktop v4

- Recall Penny desktop v3
- **Problem**
  - Worker thread updates GUI widgets
  - Only GUI thread should update GUI widgets
- **Solution**
  - Worker thread asks GUI thread to update GUI widgets

# Penny Desktop v4

- Penny desktop v4
  - See **PennySwing4** app

```
PennySwing4
  pom.xml
  runclient
  runclient.bat
  src
    main
      java
        edu
          princeton
            penny
              PennyDesktop.java
              PennyDesktopRunnable.java
              WorkerThread.java
```



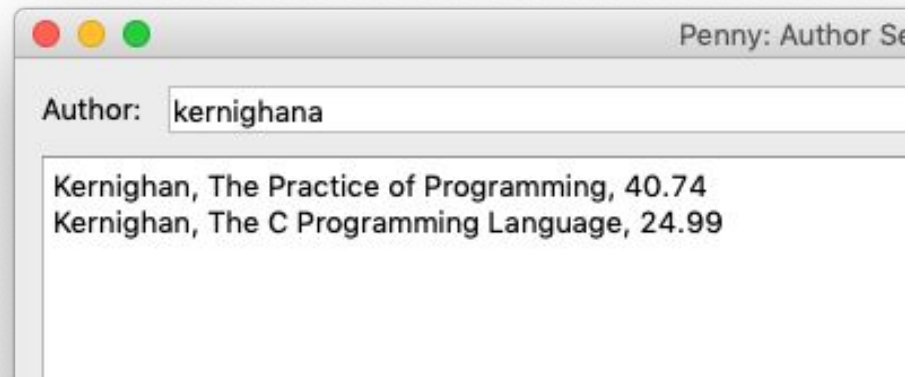
# Agenda

- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - **Version 5: Stop**
  - Version 6: Debounce

# Penny Desktop v5

- Recall Penny desktop v4
- **Problem:**
  - Server will respond to requests in arbitrary order

old  
screen  
shot



- **Solution:**
  - Abort previous request

# Penny Desktop v5

- Penny desktop v5 (cont.)
  - See **PennySwing5** app

```
PennySwing5
  pom.xml
  runclient
  runclient.bat
  src
    main
      java
        edu
          princeton
            penny
              PennyDesktop.java
              PennyDesktopRunnable.java
              WorkerThread.java
```

# Agenda

- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - **Version 6: Debounce**

# Penny Desktop v6

- Recall Penny desktop v5
- **Problem:**
  - Server could be overwhelmed with requests
- **Solution:**
  - *Debounce* the requests
- **Bonus:**
  - Reduces (but does not eliminate) the need to abort old requests!

# Penny Desktop v6

- Penny desktop v6 (cont.)
  - See **PennySwing6** app

```
PennySwing6
  pom.xml
  runclient
  runclient.bat
  src
    main
      java
        edu
          princeton
            penny
              PennyDesktop.java
              PennyDesktopRunnable.java
              WorkerThread.java
```

# Commentary

- Note:
  - The web application delivers JSON (instead of HTML), so the client reasonably can be:
    - A browser (running HTML/JavaScript code delivered by the server)
    - A desktop client

# Lecture Summary

- In this lecture we covered:
  - Desktop programming
  - A Penny **desktop** client
    - A desktop client that works with the Penny server
- See also:
  - **Appendix 1: Java Multithreaded Programming**



# Appendix 1:

# Java Multithreaded Programming

# Java Multithreaded Programming

- Recall race.py

```
$ python race.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python race.py
1
2
3
4
-1
5
6
-3
-5
-7
-9
7
8
9
10
Final balance: 10
$
```

```
$ python race.py
1
2
3
4
5
6
7
8
9
6
4
10
2
0
-2
Final balance: -2
$
```

# Java Multithreaded Programming

- See **Race.java**

```
$ java Race
1
2
3
4
5
6
7
8
0
-2
-4
-6
-8
9
10
Final balance: 10
$
```

```
$ java Race
1
2
3
4
5
6
7
8
0
-2
-4
9
10
-6
-8
Final balance: -10
$
```

# Java Multithreaded Programming

- Recall locking.py

```
$ python locking.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python locking.py
1
2
3
4
2
0
-2
-4
-6
-5
-4
-3
-2
-1
0
Final balance: 0
$
```

# Java Multithreaded Programming

- See **Locking.java**

```
$ java Locking
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ java Locking
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

# Java Multithreaded Programming

- Recall **conditions.py**

```
$ python conditions.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python conditions.py
1
2
3
4
5
3
1
2
3
4
5
6
4
2
0
Final balance: 0
$
```

# Java Multithreaded Programming

- See **Conditions.java**

```
$ java Conditions
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

```
8
```

```
6
```

```
4
```

```
2
```

```
0
```

```
Final balance: 10
```

```
$
```

```
$ java Conditions
```

```
1
```

```
2
```

```
3
```

```
4
```

```
2
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
4
```

```
2
```

```
0
```

```
Final balance: 0
```

```
$
```

# Java Multithreaded Programming

- Multithreaded programming across languages

|            | Language Support | Library Support |
|------------|------------------|-----------------|
| Python     | no               | yes             |
| C          | no               | yes             |
| Java       | yes              | yes             |
| JavaScript | no               | no              |