LEC 03

CSE 122

File I/O

Questions during Class?

Raise hand or send here

sli.do #cse122

BEFORE WE START

Slido input & discuss with neighbors:

What's the last show you watched?

Music: 122 26sp Lecture Jams 🎧

Instructor: Elba Garza

TAs:

David	Caleb	Cole	Yang
William	Neha	Blake R.	Cady
Dani	Wesley	Carson	Diya
Rohan	Isis	Sushma	
Andrew	Colin	Connor	
Ava	Naomi	Mahima	
Shreyank	Hanna	Nicolae	
Nicole	Blake P.	Ivory	

Lecture Outline (1/4)

- Announcements/Reminders
  - Review Java
- Review Scanners for User Input
- Scanners for Files
  - Token-based, Line-based, and Hybrid processing
- Example

Announcements (1/2)

- The IPL is open!
  - MGH 334, schedule is on the course website; staffed by our awesome TAs!
  - Open 12:30 to 9:30pm most weekdays, but check the schedule…
- Creative Project 0 due Thursday, April 9th at 11:59pm PT
  - Make sure to complete the "Final Submission" slide and **submit**!
  - Submit as many times as you'd like—we will only grade the latest submission made before the deadline
- Just joined CSE 122? That's okay; look at Ed & course website and catch up!
  - Freaking out that C0 is due tomorrow? It's ok! Resubmission cycles allow you to submit it later.
- Elba OH Update: Mondays 2:30–3:30pm & Thursdays 3:30–4:30pm
- Quiz dates:

Quiz 0: April 16th **Quiz 1**: May 7th **Quiz 2**: May 26th

Announcements (2/2)

- CSE 390HA Honors Seminar
  - Weekly seminar, Tuesdays at 3:30pm
  - No homework! Just show up and chat.
  - Talk about sociotechnical topics in computing, different every quarter!
    - Technology & the Environment
    - Anonymity & Privacy in Tech
    - Warfare Technology
    - Real-life Effects of Software Bugs
    - Sign up via this Google Form by end-of-week!

Reminders: Review Java Syntax

Java Tutorial reviews all the relevant programming features you should familiar with (even if you don't know them in Java).

- Printing and comments
- Variables, types, expressions
- Conditionals (if/else if/ else)
- Loops (for and while)
- Strings
- Methods
- Arrays & 2D arrays

Recordings of both sessions are now available on the course website in the course calendar!

Lecture Outline (2/4)

- Announcements/Reminders
  - Review Java
- **Review Scanners for User Input**
- Scanners for Files
  - Token-based, Line-based, and Hybrid processing
- Example

(Review) Scanner for User Input

Scanner is defined in the `java.util` package

```
import java.util.*;
```

```
Scanner console = new Scanner(System.in);
```

Scanner Methods	Description
`nextInt()`	Reads the next token from the user as an `int` and returns it
`nextDouble()`	Reads the next token from the user as a `double` and returns it
`next()`	Reads the next token from the user as a `String` and returns it
`nextLine()`	Reads an *entire line* from the user as a `String` and returns it
`hasNextInt()`	Returns `true` if the next token can be read as an `int`, `false` otherwise
`hasNextDouble()`	Returns `true` if the next token can be read as a `double`, `false` otherwise
`hasNext()`	Returns `true` if there is another token of input to be read in, `false` otherwise
`hasNextLine()`	Returns `true` if there is another line of input to be read in, `false` otherwise

Lecture Outline (3/4)

- Announcements/Reminders
  - Review Java
- Review Scanners for User Input
- **Scanners for Files** 
  - Token-based, Line-based, and Hybrid processing
- Example

(PCM) Tokens

Tokens are units of input (as defined by the Scanner) that are separated by ***whitespace*** (spaces, tabs, new lines)

Practice : Think

sli.do #cse122

How many tokens are in the following line? (1/8)

```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (2/8)

```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (3/8)

```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (4/8)

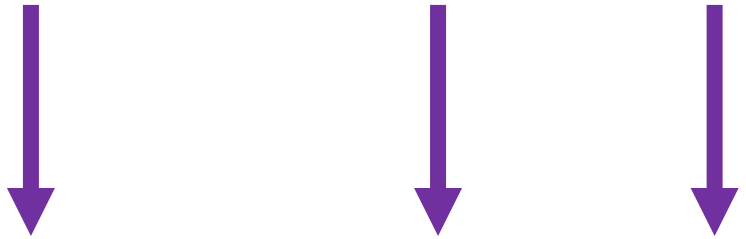
```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (5/8)



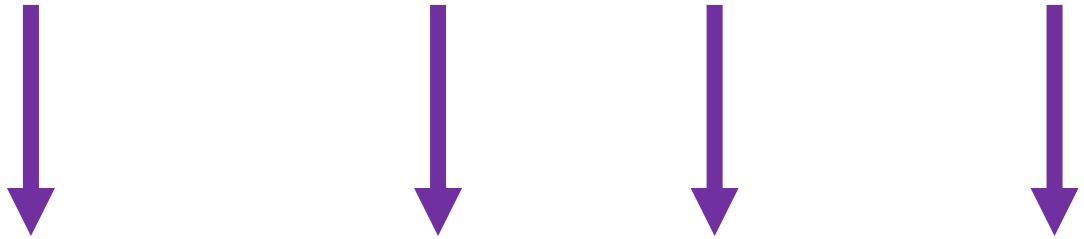
```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (6/8)



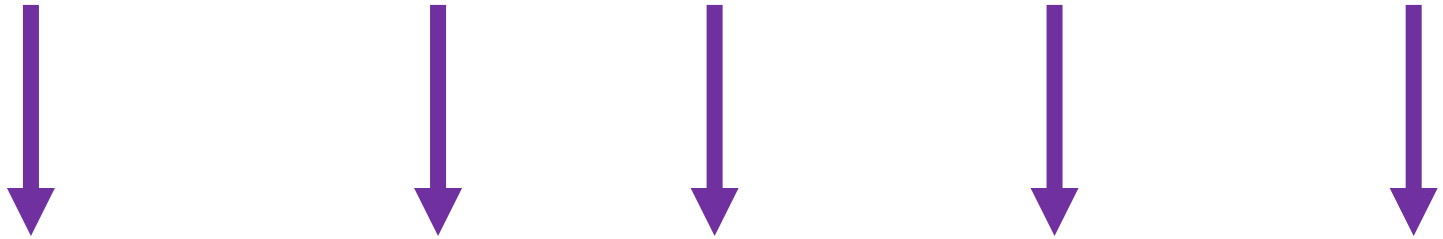
```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (7/8)



```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

Practice : Pair

sli.do #cse122

How many tokens are in the following line? (8/8)

```
“Hello world !” my-name is Elba
```

A) Four **B)** Five **C)** Six **D)** Seven

(PCM) Scanner for File I/O

`Scanner` is defined in the `java.util` package

```
import java.util.*;
```

`File` is defined in the `java.io` package

```
import java.io.*;
```

```
File file = new File("Example.txt");
Scanner fileScan = new Scanner(file);
```

Scanner Methods	Description
`nextInt()`	Reads the next token as an `int` and returns it
`nextDouble()`	Reads the next token as a `double` and returns it
`next()`	Reads the next token as a `String` and returns it
`nextLine()`	Reads an *entire line* as a `String` and returns it
`hasNextInt()`	Returns `true` if the next token can be read as an `int`, `false` otherwise
`hasNextDouble()`	Returns `true` if the next token can be read as a `double`, `false` otherwise
`hasNext()`	Returns `true` if there is another token of input to be read in, `false` otherwise
`hasNextLine()`	Returns `true` if there is another line of input to be read in, `false` otherwise

(PCM) Checked Exceptions

If you try to compile a program working with file scanners, you may encounter this error message:

```
error: unreported exception FileNotFoundException; must be
caught or declared to be thrown
```

To resolve this, you need to include `throws FileNotFoundException` at the end of the header of any method containing file Scanner creation code, or any method that calls that method!

- This is like signing a waiver and telling Java—"Hey, I hereby promise to not get mad at you when you bug and crash my program if I give you a file that doesn't actually exist."

(PCM) Typical Line Processing Pattern

```
while (scan.hasNextLine()) {
    String nextLine = scan.nextLine();
    // do something with nextLine
}
```

(PCM) Typical Token Processing Pattern (1/2)

```
while (scan.hasNext_____()) {
    _______ nextToken = scan.next_____();
    // do something with nextToken
}
```

(PCM) Typical Token Processing Pattern (2/2)

```
while (scan.hasNext_____()) {
    _______ nextToken = scan.next_____();
    // do something with nextToken
}
```

Avoid mixing line- and token-based processing!

(PCM) Typical Hybrid Pattern (1/2)

```
while (fileScan.hasNextLine()) {
    String line = fileScan.nextLine();
    Scanner lineScan = new Scanner(line);
    while (lineScan.hasNext__()) {
        __ nextToken = lineScan.next__();
        // do something with nextToken
    }
}
```

(PCM) Typical Hybrid Pattern (2/2)

```
while (fileScan.hasNextLine()) {
    String line = fileScan.nextLine();
    Scanner lineScan = new Scanner(line);
    while (lineScan.hasNext__()) {
        __ nextToken = lineScan.next__();
        // do something with nextToken
    }
}
```

Line-by-line and **then** token-by-token to do something

(PCM) Scanners with Strings (not files!) (1/7)

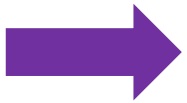
```
String str = "A quick,  brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext____()) {
    ____ nextToken = stringScan.next____();
    // do something with nextToken
}
```

(PCM) Scanners with Strings (not files!) (2/7)

```
String str = "A quick,  brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext()) {
    String nextToken = stringScan.next();
    System.out.println(nextToken);
}
```

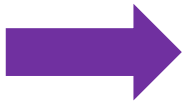
(PCM) Scanners with Strings (not files!) (3/7)

```
String str = "A quick,  brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext()) {
    String nextToken = stringScan.next();
    System.out.println(nextToken);
}
```



(PCM) Scanners with Strings (not files!) (4/7)

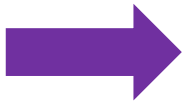
```
String str = "A quick,  brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext()) {
    String nextToken = stringScan.next();
    System.out.println(nextToken);
}
```



A

(PCM) Scanners with Strings (not files!) (5/7)

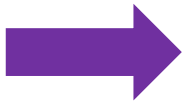
```
String str = "A quick,   brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext()) {
    String nextToken = stringScan.next();
    System.out.println(nextToken);
}
```



quick,

(PCM) Scanners with Strings (not files!) (6/7)

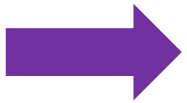
```
String str = "A quick,  brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext()) {
    String nextToken = stringScan.next();
    System.out.println(nextToken);
}
```



brown

(PCM) Scanners with Strings (not files!) (7/7)

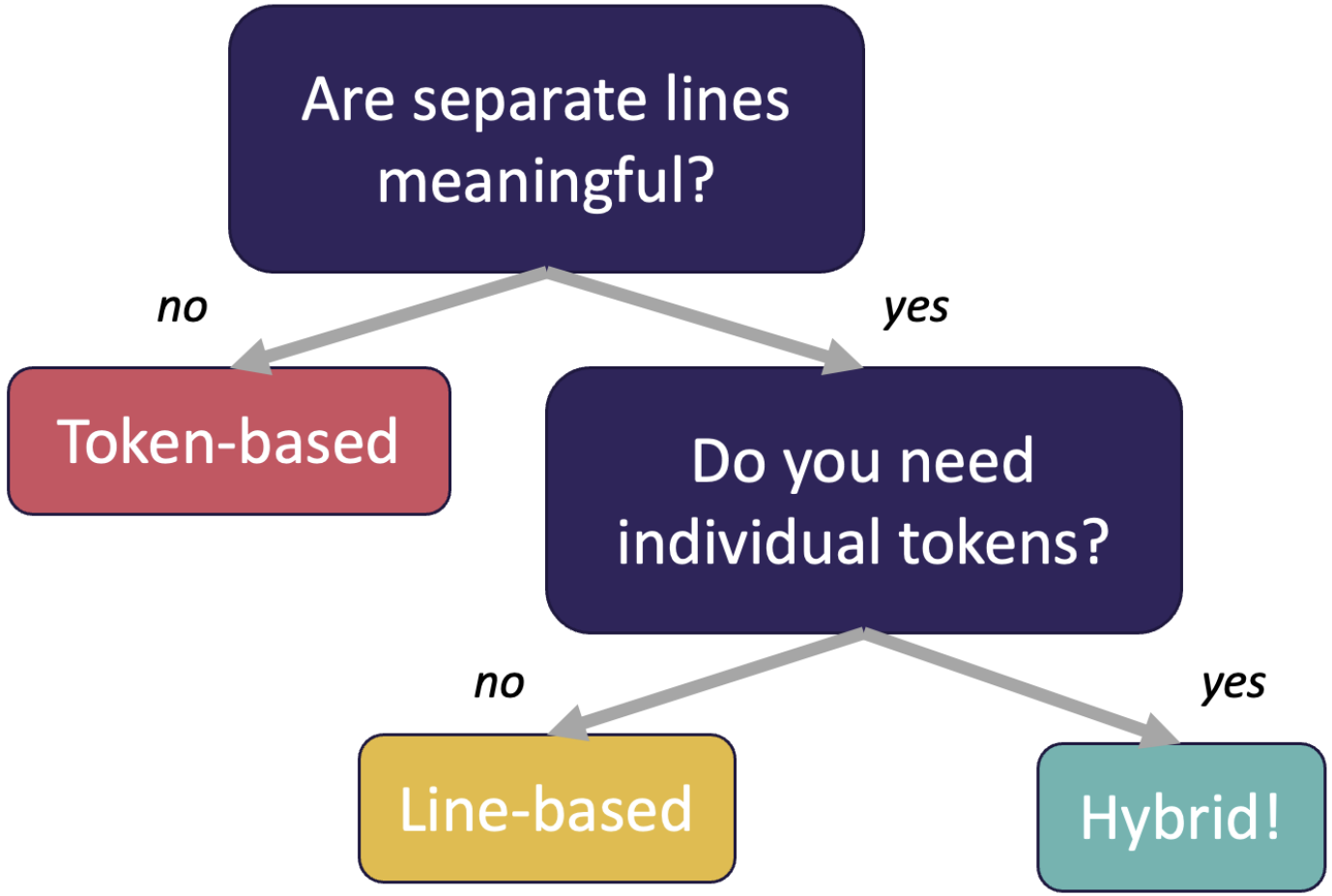
```
String str = "A quick,  brown fox";

Scanner stringScan = new Scanner(str);
while (stringScan.hasNext()) {
    String nextToken = stringScan.next();
    System.out.println(nextToken);
}
```



fox

(PCM) Token vs. Line vs. Hybrid?

- We now know 3 different ways to use Files!
  - Line
  - Token
  - Hybrid
- Although this gives us flexibility – it can sometimes get confusing
- Feel free to use the following diagram to help!



Practice : Think

sli.do #cse122

Think: What is the output of this Java method?

```
File f = new File("Example.txt");
Scanner fileScan = new Scanner(f);
while (fileScan.hasNextLine()) {
   String line = fileScan.nextLine();
   Scanner lineScan = new Scanner(line);
   while (lineScan.hasNext()) {
      System.out.print(lineScan.next() + ", ");
   }
   System.out.println();
}
```

Example.txt:

```
One Two
Three
```

A) One, Two, Three,

B) One,
Two,
Three,

C) One, Two,
Three,

D) Error / Exception

Practice : Pair

sli.do #cse122

Pair: What is the output of this Java method?

```
File f = new File("Example.txt");
Scanner fileScan = new Scanner(f);
while (fileScan.hasNextLine()) {
   String line = fileScan.nextLine();
   Scanner lineScan = new Scanner(line);
   while (lineScan.hasNext()) {
      System.out.print(lineScan.next() + ", ");
   }
   System.out.println();
}
```

Example.txt:

```
One Two
Three
```

A) One, Two, Three,

B) One,
Two,
Three,

C) One, Two,
Three,

D) Error / Exception

Initial Letter Count by Line (1/2)

- Initial Letter Count
  - Token-based processing!
- Letter Count
  - Line-based processing!
- **Task: Initial Letter Count *by Line***
  - What kind of processing?

Initial Letter Count by Line (2/2)

- Initial Letter Count
  - Token-based processing!
- Letter Count
  - Line-based processing!
- **Task: Initial Letter Count *by Line***
  - What kind of processing?
  - Hybrid processing! Each line, and then each token per line.

Lecture Outline (4/4)

- Announcements/Reminders
  - Review Java
- Review Scanners for User Input
- Scanners for Files
  - Token-based, Line-based, and Hybrid processing
- **Example** 

Movie Ratings (1/2)

In this program, we'll be examining and printing data from a file of IMDb ratings for popular U.S. movies. This will happen through 3 major user-entered commands:

(F)ind movie, **(A)dd** a rating, and **(P)rint** ratings.

small.tsv

```
5
Title   Average Total
Pride_&_Prejudice   7.8 323647
Barbie  6.9 455488
Oppenheimer 8.4       588723
Poor_Things 8.5       20542
Spider-Man:_Across_the_Spider-Verse 8.6     329200
```

Movie Ratings (2/2)

```
Welcome to the CSE 122 Movie Rating Program!
Loaded 5 movies from small.tsv!

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: F
What's the name of the movie? Pride_&_Prejudice
Movie Pride_&_Prejudice found!
    Average Rating: 7.8
    Total Ratings: 323647

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: A
What movie would you like to add your rating to? Pride_&_Prejudice
And what rating would you like to give? 100000

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: f
What's the name of the movie? Pride_&_Prejudice
Movie Pride_&_Prejudice found!
    Average Rating: 8.1
    Total Ratings: 323648

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: P
5
Title Average Total
Pride_&_Prejudice 7.8 323647
Barbie 6.9 455488
Oppenheimer 8.4 588723
Poor_Things 8.5 20542
Spider-Man:_Across_the_Spider-Verse 8.6 329200

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: q
Thank you for using this program! Bye!
```

small.tsv

```
5
Title   Average Total
Pride_&_Prejudice  7.8 323647
Barbie  6.9 455488
Oppenheimer 8.4  588723
Poor_Things 8.5  20542
Spider-Man:_Across_the_Spider-Verse 8.6  329200
```

Movie Ratings: Development Strategy

1. Fill in the main method with a loop that calls a method to read the data in from the .tsv file and allows the user to select between the different options (find, add, print, quit). I was nice and did this for you 😇
2. Implement a method to **load** the movie rating data from the file and populate the appropriate arrays.
3. Implement a method that allows users to **find** the rating for a movie.
4. Implement a method that allows users to **add** a rating for a movie.
5. Implement a method that allows users to **print** the movie rating data. I was nice and did this for you 😇