

CSE 121 – Lesson 15

Miya Natsuhara

Spring 2023

Music: [121 23sp Lecture Vibes](#) 




[sli.do #cse121](https://sli.do/#cse121)

TAs:

<i>Jasmine</i>	<i>Atharva</i>	<i>Mia</i>	<i>Justin</i>
<i>Shananda</i>	<i>Julia</i>	<i>Archit</i>	<i>Aishah</i>
<i>Vidhi</i>	<i>Anju</i>	<i>Grace</i>	<i>Claire</i>
<i>Larry</i>	<i>Lydia</i>	<i>Kailye</i>	<i>Lydia</i>
<i>Jacqueline</i>	<i>Jonus</i>	<i>Joshua</i>	<i>Kai</i>
<i>Afifah</i>	<i>Hugh</i>	<i>James</i>	

Announcements, Reminders

- Form for Quiz Retakes on May 23 released later today
 - Quiz 1 and Quiz 2 will be eligible for retake **(last opportunity for Quiz 1)**
- Quiz 2 grades posted later today
- Creative Project 2 was released on Wednesday
 - File I/O and last time with Turtles 
 - Due Tuesday **May 23**
- Reminder: **Final exam Thursday, June 8 2:30pm-4:20pm**

(PCM) Scanner & File for File I/O

File is defined in the `java.io` package

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

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

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

(PCM) Typical Line-Processing Pattern

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

(PCM) Typical Token-Processing Pattern

```
while (fileScan.hasNext__()) {  
    __ nextToken = fileScan.next__();  
    // do something line nextToken  
}
```

(PCM) Typical Hybrid Pattern

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

(PCM) Scanning Numeric Data

On Wednesday, we primarily used `String`-based Scanner methods to read input from a file. Let's work with some numeric data now!

We're going to make more use of

- `hasNextInt()`
- `hasNextDouble()`
- `nextInt()`
- `nextDouble()`
- Assumptions about our file's format!

Poll in with your answer!



What would be the result of running the FindMinAndMax program with this as input?

```
2.3 9.2
    17
3.14 4.83 0.73
          -1.0005
```

- A. Error
- B. minimum was -1.0005 and maximum was 17.0
- C. minimum was 0.73 and maximum was 17
- D. minimum was 0.73 and maximum was 17.0

Poll in with your answer!



What would be the result of running the FindMinAndMax program with this as input?

2.3 9.2
17 0.73

- A. Error
- B. minimum was 0.0 and maximum was 17.0
- C. minimum was 0.73 and maximum was 17.0
- D. minimum was 0.73 and maximum was 17

(PCM) PrintStream

PrintStream is defined in the `java.io` package

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

```
File outputFile = new File("out.txt");
```

```
PrintStream output = new PrintStream(outputFile);
```

Scanner Methods	Description
<code>print(...)</code>	Prints the given value to the set output location.
<code>println(...)</code>	Prints the given value to the set output location, and then terminates the line.