CSE 121 – Lesson 14

Miya Natsuhara
Spring 2023

Music: 121 23sp Lecture Vibes 🌸

TAs:
- Jasmine
- Shananda
- Vidhi
- Larry
- Jacqueline
- Afifah
- Mia
- Atharva
- Julia
- Anju
- Lydia
- Jonus
- Hugh
- Archit
- Grace
- Kailye
- Joshua
- James
- Justin
- Aishah
- Claire
- Lydia
- Kai

sli.do #cse121
Announcements, Reminders

• Quiz 2 tomorrow (Thursday, May 18) in quiz sections
  • While loops, user input

• Quiz Retakes – May 23
  • Form will be posted soon
  • Quiz 1 and Quiz 2 will be eligible for retake (last opportunity for Quiz 1)

• Creative Project 2 will be released later tonight
  • Due Tuesday May 23

• Study Participation Opportunity!

• Reminder: Final exam Thursday, June 8 2:30pm-4:20pm
Formative Feedback: Closing the Loop

The Good...

• Resubmission and Retake Opportunities
• Pre-Class Work
• In-Class Demos
• Quiz Sections
• IPL
Formative Feedback: Closing the Loop

Suggestions...

• More practice problems for Quizzes
• Assignment specification improvements
• Gradebook
Formative Feedback: Closing the Loop

Reminders...

• Practice Problems & Resources tab
• Slides from Quiz Section
• Resubmission Deadlines
(PCM) Scanner & File for File I/O

File is defined in the java.io package
import java.io.*;

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

File newFile = new File("newFile.txt");
Scanner fileScan = new Scanner(newFile);
(PCM) Checked Exceptions

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

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

To resolve this, you need to be `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 out and crash my program if I give you a file that doesn't actually exist."
Typical Line-Processing Pattern

```java
while (fileScan.hasNextLine()) {
    String line = fileScan.nextLine();
    // do something with line
}
```
while (fileScan.hasNext()) {
    __ nextToken = fileScan.next__();
    // do something line nextToken
}
(PCM) Typical Hybrid Pattern

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