

CSCI 132:

Basic Data Structures and Algorithms

Java file I/O

Reese Pearsall
Spring 2025

Announcements

On Friday we will be discussing
Program 1



Public property



Private property
with public getter
and setter

ProgrammerHumor.io

Reading in a file in Java

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

Opening a reading a file must be done inside of a try/catch statement

Reading in a file in Java

Open a file with the
FileReader library

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

Reading in a file in Java

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

Open a file with the
FileReader library

Read contents of file
with Scanner

Reading in a file in Java

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

Open a file with the
FileReader library

Read contents of file
with Scanner

Read headers of csv
file (discard them)

Reading in a file in Java

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

Open a file with the
FileReader library

Read contents of file
with Scanner

Read headers of csv
file (discard them)

Iterate through each
line of file

Reading in a file in Java

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

Open a file with the
FileReader library

Read contents of file
with Scanner

Read headers of csv
file (discard them)

Iterate through each
line of file

Get each line and print
it out

Reading in a file in Java

```
FileReader file;  
try {  
    file = new FileReader("video_games.csv");  
  
    Scanner inFile = new Scanner(file);  
  
    inFile.nextLine();  
  
    while(inFile.hasNext()) {  
        String line = inFile.nextLine();  
        System.out.println(line);  
  
        String[] splitted_line = line.split(",");  
        System.out.println(splitted_line[0]);  
    }  
}
```

Open a file with the
FileReader library

Read contents of file
with Scanner

Read headers of csv
file (discard them)

Iterate through each
line of file

Get each line and print
it out

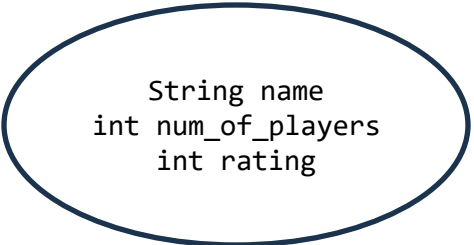
To print out the first column of the
file, we can split each line, and print
out the first column

"Reese, Susan, Spencer" *One long String*
↓
.split(",") → ["Reese", "Susan", "Spencer"]
Array of Strings

Writing to a file in Java

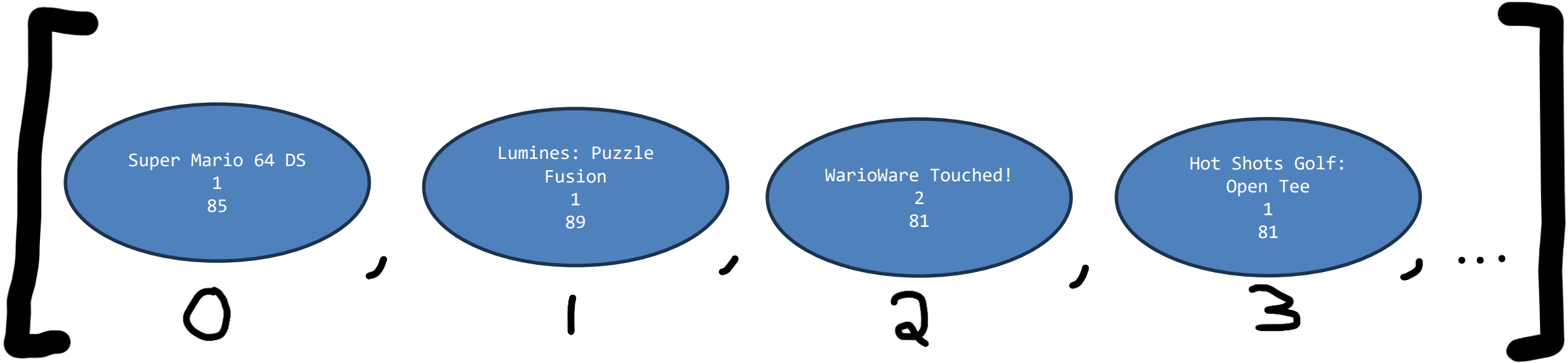
```
try {  
    BufferedWriter writer = new BufferedWriter(new FileWriter("output.txt"));  
    writer.write("Hello World! \n");  
    writer.write("-----\n");  
    writer.close();  
} catch (IOException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

VideoGame Class Blueprint



	A	B	C	D	E	F	G	H
1	Title	Features.F	Features.Max Players	Features.F	Features.C	Metadata.	Metadata.	Metrics.Review Score
2	Super Mario 64 DS	TRUE	1	TRUE	TRUE	TRUE	TRUE	85
3	Lumines: Puzzle Fusion	TRUE	1	TRUE	TRUE	TRUE	TRUE	89
4	WarioWare Touched!	TRUE	2	TRUE	TRUE	TRUE	TRUE	81
5	Hot Shots Golf: Open Tee	TRUE	1	TRUE	TRUE	TRUE	TRUE	81
6	Spider-Man 2	TRUE	1	TRUE	TRUE	TRUE	TRUE	61
7	The Urbz: Sims in the City	TRUE	1	TRUE	TRUE	TRUE	TRUE	67
8	Ridge Racer	TRUE	1	TRUE	TRUE	TRUE	TRUE	88
9	Metal Gear Ac!d	TRUE	1	TRUE	TRUE	TRUE	TRUE	75
10	Madden NFL 2005	TRUE	1	TRUE	TRUE	TRUE	TRUE	68
11	Pokmon Dash	TRUE	1	TRUE	TRUE	TRUE	TRUE	46

VideoGame[] array



Writing to a file in Java

```
try {  
    BufferedWriter writer = new BufferedWriter(new FileWriter("output.txt"));  
    writer.write("Hello World! \n");  
    writer.write("-----\n");  
    writer.close();  
} catch (IOException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

Name of output file

Writing to a file in Java

```
try {  
    BufferedWriter writer = new BufferedWriter(new FileWriter("output.txt"));  
    writer.write("Hello World! \n");  
    writer.write("-----\n");  
    writer.close();  
} catch (IOException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

We write out contents to file using the `write()` method

Writing to a file in Java

```
try {  
    BufferedWriter writer = new BufferedWriter(new FileWriter("output.txt"));  
    writer.write("Hello World! \n");  
    writer.write("-----\n");  
    writer.close();  
} catch (IOException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

When writing to a file, it is very important to close the file when you are finished