



CS 2110, SP24

# Discussion 4: File input & output

# Mini-review: Input and output

References:

- Textbook Supplement 2
- Website reading

# File paths

- Files have a **filename** on your computer's filesystem
  - Example: **Main.java**
- Files live in folders; the list of folders, separated by '/', is the file's **path**
  - Example: **/home/bob/cs2110/a3/Main.java**
- Absolute paths start at the filesystem "root"
  - Examples: **C:/** (Windows; yes, / can be used instead of \), **/** (Mac/Linux)
- Relative paths assume you start from a particular folder
  - Applications are run from a "current directory" (when running from inside IDEA, this is your project's folder)
  - Example: **cs2110/a3/Main.java** (no leading slash) when current directory is **/home/bob**

# Demo: Open a text file in an IDEA project

```
import java.io.Reader;  
import java.io.FileReader;
```

```
Reader in = new FileReader("hello.txt");
```

## FileReader

```
public FileReader(String fileName)  
    throws FileNotFoundException
```

Creates a new `FileReader`, given the name of the file to read, using the default charset.

### Parameters:

`fileName` - the name of the file to read

### Throws:

`FileNotFoundException` - if the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading.

### See Also:

`Charset.defaultCharset()`

# Recall

What are your two options when you call a method that might throw a checked exception?

- 1.

- 1.

# Recall

What are your two options when you call a method that might throw a checked exception?

1. Catch the exception (**try** / **catch**)
  - “I can handle this”
1. Propagate the exception (**throws**)
  - “I can’t do my job without this; better tell my supervisor”

# Demo: Read lines and tokens

```
Reader in = new FileReader(path);
Scanner lines = new Scanner(in);
while (lines.hasNextLine()) {
    String line = lines.nextLine();
    // Process line
    Scanner tokens = new Scanner(line);
    tokens.useDelimiter(",");
    int firstToken = tokens.nextInt();
}
```

# Indexed tokens: `String.split()`

- Returns an array of tokens separated by argument

```
String phone = "607-555-5309";  
String[] tokens = phone.split("-");  
String areaCode = tokens[0];  
String exchange = tokens[1];  
String lineNum = tokens[2];
```