

File I/O

Definition: The input and output operations that a program can perform on a file.

Reading and Writing Text Files

- `File inFile = new File("input.txt")`
`Scanner in = new Scanner(inFile);`
 - Allows the `Scanner` object to read text from the `input.txt` file
- `PrintWriter out = new PrintWriter("output.txt");`
 - Allows the `PrintWriter` object to write text to the `output.txt` file
 - Any information in an existing file will be deleted before new data is added
 - Creates an empty file if the file does not exist
 - Make sure you close the `PrintWriter` object (`out.close();`), or the disk file may not have all of the output
- `FileNotFoundException`
 - Can occur when the input or output file does not exist
 - A compile-time error occurs if this is not included, which makes it different from other exceptions

Common Error

- File names must be written in code with two backslashes if you want the backslash to be included
 - `inFile = new File("c:\\homework\\input.dat")`
 - The first backslash is an escape character

File Dialog Boxes

- GUI's for opening and saving files
 - `JFileChooser chooser = new JFileChooser();`
`Scanner in = null;`
`if(chooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {`
 `File selectedFile = chooser.getSelectedFile();`
 `in = new Scanner(selectedFile);`
 ...
`}`
 - `Null` can be replaced with a component, and the dialogue box will appear over the component
 - The `APPROVE_OPTION` needs to be tested because the user may cancel

Reading Web Pages

- `String address = "http://java.sun.com/index.html";`
`URL locator = new URL(address);`
`Scanner in = new Scanner(locator.openStream());`
 - Beware: An `IOException` may occur with the `URL` constructor and `openStream()` method

Reading Text Input

- To separate words based on any characters other than letters, use: `in.useDelimiter("[^A-Za-z]+");`
- To separate words based on numbers, use: `in.useDelimiter("[0-9]+");`
- Processing Lines: `nextLine()`
 - Useful when programmers need two or more words for a name
- Reading Numbers: `nextInt()` and `nextDouble()`
 - A `NoSuchElementException` can occur if there are no numbers
 - An `InputMismatchException` can occur if inappropriate letters are mixed with digits
 - Use `hasNextInt()` and `hasNextDouble()` to avoid the exceptions
 - Beware: These methods may leave empty lines after retrieving the integer or double values
- Reading Characters: `hasNext()` and `next()`
 - Useful when individual characters are important to a program
 - Use: `in.useDelimiter("");`; so that `next()` will return a string of a single character