**Lesson 3 – Files**

* When using files, we must import the java.io package
* We also need to import the java.util package (to get the Scanner package)
* When working with files we must follow a **3-step process**
  + **Open the File**
  + **Input / Output data**
  + **Close the file**
* **Open the File**
  + Create a File Object
    - *File file = new File (String filename);*
  + Then pass the File Object into the Scanner’s constructor
    - *Scanner input = new Scanner(file);*
  + The program will crash if the file is not found
    - Prevent a crash by using a Try-Catch statement and throw a FileNotFoundException
  + To ensure a file exists the instantiation could be placed in a loop that runs until the file is not found
    - We can give the user more information on why an error is occurring using **File Methods**
      * E.g isDirectory(), exists(), canRead(), etc.
* **Input (Line by Line) using Scanner**
  + **Uses the following Scanner Methods**
    - Boolean hasNextLine()
      * Determines if there is another line in the file
    - String nextLine()
      * Returns all remaining text in the current line of the file and sets the cursor to the beginning of the next line in the file
    - String next()
      * Returns the next Token
        + Skipping any whitespace
    - int nextInt(), double nextDouble(), etc
      * Returns the next Token as the specified type
        + Skipping any whitespace
        + Throws InputMismatchException if it is not the right type
  + **Delimiters (Helps prevent errors / bugs)**
    - Characters that separate one Token from the next
    - The Default Delimiters are whitespace characters
      * Space, tab, endOfLine
    - If a data item has a space as part of the data, it can mess up the input algorithm
    - Instead, data items can be separated by a different delimiter
      * Like “:”
    - A Delimiter can be setup with the following code
      * *ScannerObject.useDelimiter (String delimiter);*
* **Output**
  + To output to a file we need to create a PrintWriter Object that takes the filename as its parameter
  + Must close a file when we are done outputting data
    - Files take up a lot of resources so its important to close them
    - Data will not be saved in the file we are outputting to if it is not closed