AP CS: Lesson 13: File Input

Name: ___________________________ Period: ___

Java Syntax:

Scanner & File Objects:

Required Library for these: (placed above the Class creation)

```
        import java.util.*;     // required for Scanner Object
        import java.io.*;       // required for File Object
```

Scanner & File Object Creations:

```
        Scanner name = new Scanner(source);
        File name = new File("file name");
```

Examples:

```
        Scanner console = new Scanner(System.in); // creates Scanner named “console”
                                            // that reads from the input (keyboard)
        File fileHere = new File("mydata.txt");   // creates File named “fileHere” that
                                                  // accesses the file mydata.txt
        Scanner input = new Scanner(fileHere);    // creates Scanner named “input” that
                    // reads from the file “fileHere”, which is accessing “mydata.txt”
        // Or can be done in one line shorter like this:
        Scanner input = new Scanner(new File("mydata.txt"));  // Same as above 2 lines
```

File Methods: (f is the File object)

```
        f.delete() removes file from disk
        f.getName() returns file's name
        f.length() returns number of bytes in file
        f.renameTo(file) changes name of file
```

File Test Methods

```
        f.canRead() returns whether file is able to be read
        f.exists() whether this file exists on disk
```

throws clause: Keywords on a method's header that state that it may generate an exception (and will not handle it).

Syntax:

```
        public static type name(params) throws type {
```

Example: **and Required for Classes that read files !!**

```
        public class ReadFile {
            public static void main(String[] args)
                    throws FileNotFoundException {
```

A token: A unit of user input, separated by whitespace (blanks, tabs, & new line)

- used by all Scanner Methods except .nextLine()

Scanner Methods: (s is the Scanner object)

```
        s.nextInt()  reads an int from the user and returns it
        s.nextDouble()  reads a double from the user
        s.next()  reads a one-word String from the user
        s.nextLine()  reads a one-line String from the user
```

Scanner Test Methods

```
        s.hasNext() // returns true if there is a next token
        s.hasNextInt() // returns true if there is a next token & it can be read as int
        s.hasNextDouble() // returns true if there is a next token and it can
                          // be read as a double
        s.hasNextLine()   // returns true if there are any more lines of input to read
                          //  (always true for console input)
```

AP CS: Lesson 13: File Input

Class Notes:

Exit Ticket - Please answer and return at end of period. Thanks.

1. I understand how to prevent a “Fencepost” condition: __ Yes __ Somewhat __ No
2. I understand how to use to while loop: __ Yes __ Somewhat __ No
3. I understand how to use read/scan from a file: __ Yes __ Somewhat __ No
4. I understand how to scan & skip tokens (i.e. using hasnextInt()): __ Yes __ Somewhat __ No

Additional comments, especially if you answered “No” to any of the above what would you like to more details on? Thanks

____

____

____

____

____