CS 2334: Lab 3
Importing a CSV File

Notes

- Rubric for each lab and project tells you what we are specifically looking for when we are grading your assignments
- Don’t forget to:
  - Add documentation where appropriate
  - Run javadocs **after** your documentation is complete

Importing a File

In Lab 1, we used a BufferedReader to take input from the keyboard. BufferedReader can also be used to take input from a file. We do this by using a FileReader.

```
BufferedReader br = new BufferedReader (new FileReader("filename.txt"));
```

Note: for this to work, the file to which FileReader refers needs to be in the project folder. Also, be sure to close the input stream. This is done by using close() on the BufferedReader.

CSV Files

CSV stands for comma separated values. It is a standard way for tables to be stored in a plain text format. This format makes it easy for us to parse the data saved therein.

An Example

Blue	5	3.2	Dog
Yellow	7	9	
Green	Hello		World

The .csv of the above table would look like this:

Blue,5,3.2,Dog
Yellow,7,9
Green,Hello,,World

Notes:

- In the second line there are only 3 objects. There is no placeholder for the last blank item.
- There is placeholder in the 3rd line to denote the blank spot in the table.

String.split

When you import a .csv, you will almost always use String.split. The split method returns a String []. It is important to note that it is not restricted to a certain size.

Blue,5,3.2,Dog

Yellow,7,9

Green,Hello,,World

Test,,,,,

Using split(“,”) on the lines above would give you the following String []:

{“Blue”, “5”, “3.2”, “Dog”} Length: 4

{“Yellow”, “7”, “9”} Length: 3

{“Green”, “Hello”, “”, “World”} Length: 4

{“Test”} Length: 1

ArrayLists

Unlike an array, the length of an ArrayList can change dynamically.

Reading in a list of lines from a BufferedReader and storing in an ArrayList:

```
ArrayList<String> list = new ArrayList<String>();  // ArrayList of Strings
String strg = br.getLine();   // Read first line
while (strg != null){         // Iterate as long as there is a next line
      list.add(strg);         // Add the line to the ArrayList
      strg = br.getLine();    // Attempt to get the next
}
```

For Each Loop

```
ArrayList<String> list = new ArrayList<String>();  // ArrayList of Strings
String strg = br.getLine();   // Read first line
while (strg != null){         // Iterate as long as there is a next line
      list.add(strg);         // Add the line to the ArrayList
      strg = br.getLine();    // Attempt to get the next
}

// Iterate over the list and do something with each item (in this case print)
for(String str: list)
      System.out.println(str);
```

Eclipse Generate Getters/Setters

Once you have your class/instance variables declared, Eclipse can generate your getters and setters for you.

- Source -> Generate Getters and Setters…

Demonstration: generating getters and setters.

Lab 3 Preparation

- Download lab3.zip
- Import into your Eclipse project

(details of how to do this are in the lab specification)

Course Schedule

Demonstrate: course schedule .csv (in Excel and raw) and on website

Lab 3

- Separate the schedule.csv file into separate course events
- Store the list of events in an ArrrayList
- Generate different types of event lists

Lab 3: CourseEvent Class

Stores information for one day from the course

- Lecture or lab number
- Date
- Event description
- Readings
- Assignments
- Items due

Lab 3: CourseEvent Class

We provide a partial implementation of CourseEvent:

- Instance variables
- Constructor prototype (but not implementation)
- Full or partial implementations of other methods

Look for “TODO”

Lab 3: Driver Class

- Read in a .csv file
- Use each line to create a CourseEvent object
- Place these objects into an ArrayList<CourseEvent>
- Generate 3 reports:
  - List of all events that are lectures
  - List of all events that are labs
  - List of all events that have something due

  - See Lab3ExpectedOutput.txt file for what we are looking for

Submission

- Submit only one file: lab3.zip (casing matters)
- Due date: Friday, September 11th @11:59pm
- Submit to lab3 dropbox on D2L