Side Topic Part 2
Loading Data from Text Files

Operating on persistent lists

Previously

Reading Text Files in Java

Reading a Text File

- Create a method `openFile` that accepts a String `filename`
- You may use the following template code to read a text file:

```
Path path = Paths.get(filename);
Charset cs = StandardCharsets.US_ASCII;
// you can also use UTF_8

try (BufferedReader reader =
Files.newBufferedReader(path, cs)){
      String line = null;
      while((line = reader.readLine()) != null){
            // do something with line
      }
} catch(IOException x){
      System.err.println(x);
}
```

Menu as a Text File

<number of items>

<item 1 name>

<item 1 description>

<number of item 1 sizes>

<item 1 size 1>

…

<item 1 size n>

<item 1 price 1>

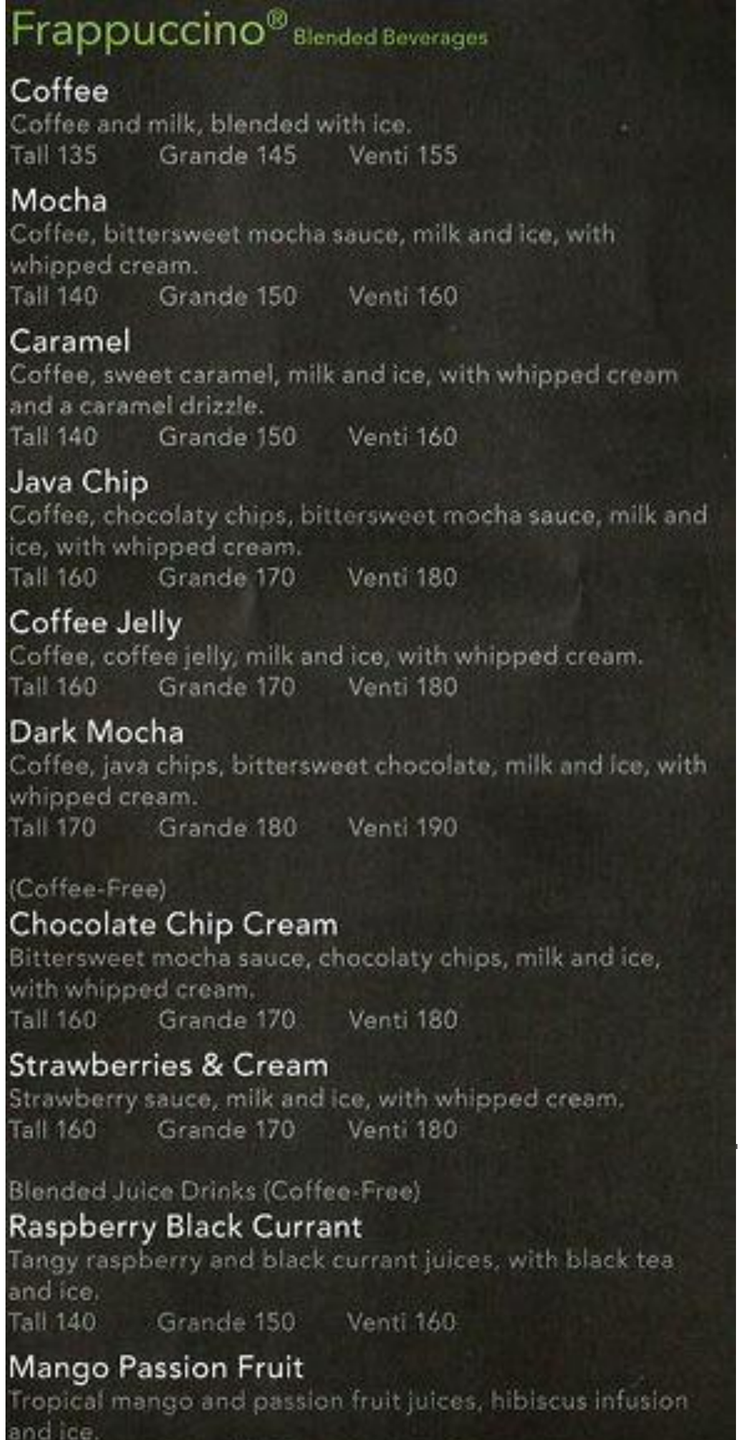
…

<item 1 price n>

…

<item n name>

…



Frappuccino® Blended Beverages

Coffee
Coffee and milk, blended with ice.
Tall 135 Grande 145 Venti 155

Mocha
Coffee, bittersweet mocha sauce, milk and ice, with whipped cream.
Tall 140 Grande 150 Venti 160

Caramel
Coffee, sweet caramel, milk and ice, with whipped cream and a caramel drizzle.
Tall 140 Grande 150 Venti 160

Java Chip
Coffee, chocolaty chips, bittersweet mocha sauce, milk and ice, with whipped cream.
Tall 160 Grande 170 Venti 180

Coffee Jelly
Coffee, coffee jelly, milk and ice, with whipped cream.
Tall 160 Grande 170 Venti 180

Dark Mocha
Coffee, java chips, bittersweet chocolate, milk and ice, with whipped cream.
Tall 170 Grande 180 Venti 190

(Coffee-Free)
Chocolate Chip Cream
Bittersweet mocha sauce, chocolaty chips, milk and ice, with whipped cream.
Tall 160 Grande 170 Venti 180

Strawberries & Cream
Strawberry sauce, milk and ice, with whipped cream.
Tall 160 Grande 170 Venti 180

Blended Juice Drinks (Coffee-Free)
Raspberry Black Currant
Tangy raspberry and black currant juices, with black tea and ice.
Tall 140 Grande 150 Venti 160

Mango Passion Fruit
Tropical mango and passion fruit juices, hibiscus infusion and ice.

…

Try something **easier**

Ask for n numbers…

Problem

Ask the user for a number **n**. Then, ask the user for **n** numbers and store all these values in int array. Print out the values in ascending order.

A familiar problem

- Normally, it would go something like this:

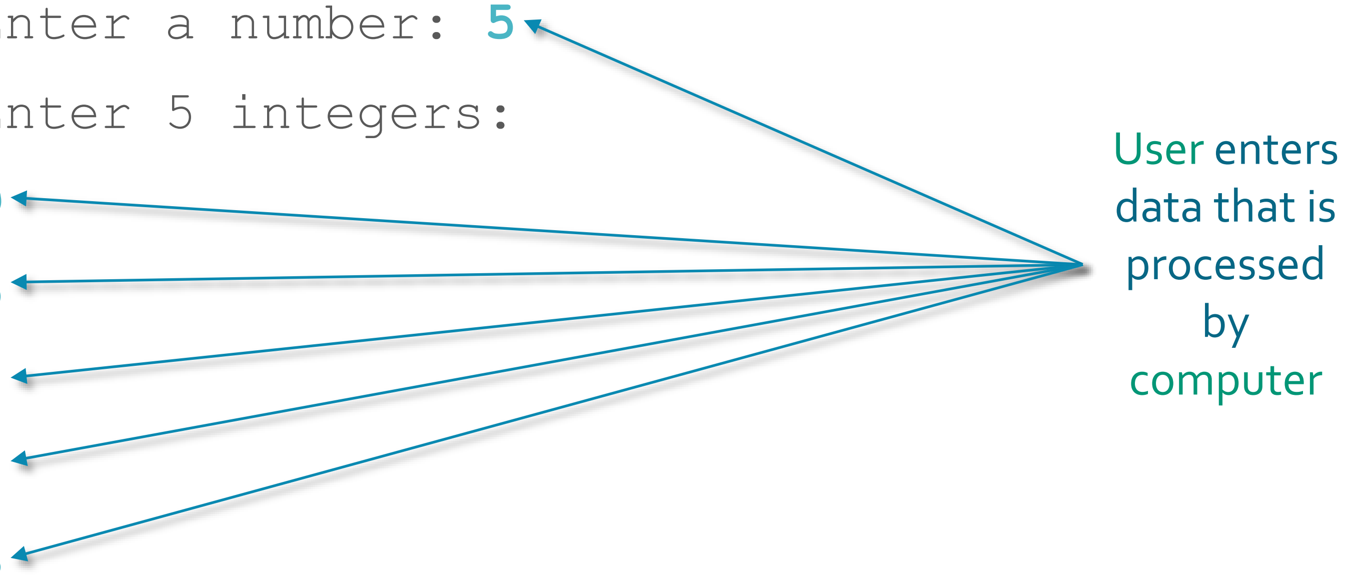
```
System.out.print("Enter a number:");
int n = sc.nextInt();
System.out.println("Enter "+n+" integers:");
int[] intArr = new int[n];
for(int i = 0; i < n; i++)
        intArr[i] = sc.nextInt();
Arrays.sort(intArr);
for(int i = 0; i < n; i++)
        System.out.print(intArr[i]+",");
```

Let's focus on ***how*** we get data

A familiar sample

- A sample of what a user would see is shown below, note that the **blue** values were those typed by the user:

```
Enter a number: 5
Enter 5 integers:
9
5
8
2
3
2,3,5,8,9,
```

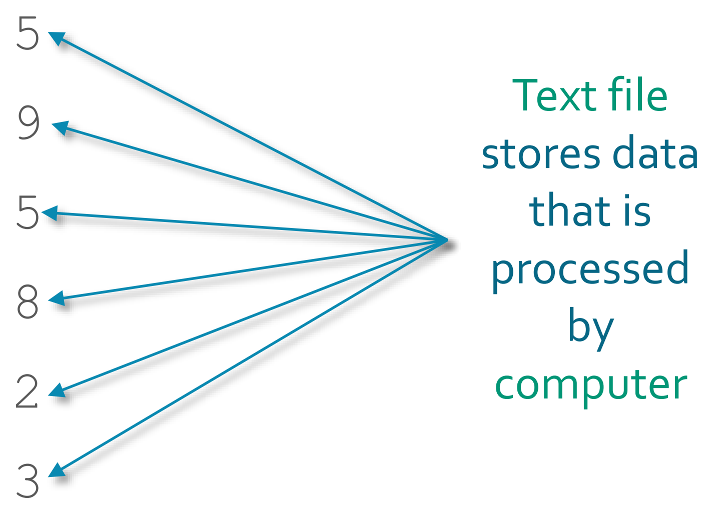


Comparing the two sources of data

In the Console

```
Enter a number: 5
Enter 5 integers:
9
5
8
2
3
```

In a Text File



```
5
9
5
8
2
3
```

Text file stores data that is processed by computer

Comparing the code that gets the data

In the Console

```
System.out.print("Enter a number:");
int n = sc.nextInt();
System.out.println("Enter "+n+" integers:");
int[] intArr = new int[n];
for(int i = 0; i < n; i++)
  intArr[i] = sc.nextInt();
```

In a Text File

```
Path path = Paths.get(filename);
Charset cs = StandardCharsets.US_ASCII;
int[] intArr = null;

try (BufferedReader reader =
Files.newBufferedReader(path, cs)){
  String line = null;
  int n = Integer.parseInt(reader.readLine());
  intArr = new int[n];
  int i = 0;
  while((line = reader.readLine()) != null && i < n){
    intArr[i] = Integer.parseInt(line);
    i++;
  }
} catch(IOException x){
  System.err.println(x);
}
```

Notice the similarities?

We get…

```
public static int[] intArr;

public static void openFile(String filename){
  Path path = Paths.get(filename);
  Charset cs = StandardCharsets.US_ASCII;

  try (BufferedReader reader = Files.newBufferedReader(path, cs)){
    String line = null;
    int n = Integer.parseInt(reader.readLine());
    intArr = new int[n];
    int i = 0;
    while((line = reader.readLine()) != null && i < n){
      intArr[i] = Integer.parseInt(line);
      i++;
    }
  } catch(IOException x){
    System.err.println(x);
  }
}
```

Checkpoint

- ✓ Create a standard formatted text file
- ✓ Read the file
- ✓ Save the data in an array
- Load the menu data into arrays

Can you make the code to load this data?

```
<number of items>
<item 1 name>
<item 1 description>
<number of item 1 sizes>
<item 1 size 1>
…
<item 1 size n>
<item 1 price 1>
…
<item 1 price n>
…
<item n name>
…
```

```
5
Coffee
Coffee and milk blended with ice
3
Tall
Grande
Venti
135
145
155
Mocha
Coffee, bittersweet mocha sauce…
…
```

Exercise 3

Go to Edmodo file section

Extra: Writing to a Text File

- Writing to a text file is actually very similar to reading

```
try (BufferedWriter writer =
Files.newBufferedWriter(path, cs)){
      for(String data : yourSourceOfData){
            writer.write(data);
            writer.newline();
      }
} catch(IOException x){
      System.err.println(x);
}
```