

# Homework 11

## Input / Output

**Do this homework for attendance at the twelfth laboratory session**

The following exercises are a tutorial on learning how to do input and output of data from files, and how to do tokenizing to extract data from strings.

For the next 4 exercises the data you will be using are stored in the `weather.txt` file. They are weather data on a US town for every day starting from the year 1950 all the way up to 2001. The weather information for one day are: date, high temp, low temp, precipitations, and snow. The temperatures are in Fahrenheit degrees. Here is the conversion formula from Fahrenheit to Celsius degrees.

$$\text{Temp}_{\text{Celsius}} = (\text{Temp}_{\text{Fahrenheit}} - 32) / 1.8$$

In order to have the `weather.txt` file download on your hard disk and unzip the attached `weather.zip` file.

### Exercise 1

Write a program that prints out the weather data for a date read from the command line. Use Celsius degrees for temperatures.

## Exercise 2

Write a program that prints out variations of high and low temperatures for the same date (mm/dd) in all years.

1. Read the date with `JOptionPane.showInputDialog()`.
2. Use a graphical representation of variations as the following:

```
1950 |-----X-----*-----|
1951 |-----X-----*-----|
1952 |-----X-----*-----|
. . . . .
2001 |-----X-----*-----|
      MinTemp                               MaxTemp
```

## Exercise 3

Extend the previous program to include a class called `DailyWeather`. Objects of the class contain the weather information (date, high temperature, low temperature, precipitations, and snow) for one day.

1. Define all the needed methods (especially `toString`) and demonstrate their efficacy by rewriting the Exercises 1 and 2 with the `DailyWeather` class.
2. Define the class constructors.

*Hint.* What do you think of a constructor like this:

```
DailyWeather(BufferedReader br) { }?
```

If you find it is useful for our exercise, define it and then use it in all the following exercises.

## Exercise 4

In the Exercise 3 did you define the data members of the `DailyWeather` class as is shown below?

```
public class DailyWeather {
    private int month;
    private int day;
    private int year;
    private int high;
    private int low;
    private int precip;
    private int snowfall;

    ...

} // DailyWeather
```

If you did, change the class definition because you ignored to abstract conceptually alike data into a single abstraction. In our example, the date (month, day, year) is a conceptual entity that is very different from the rest of the data. You should define a separate class (say `Date`) for this purpose and then use composition to define the `DailyWeather` class<sup>1</sup>. Keep all methods and operations relating to the `Date` class separate from the operations need on the weather data.

Add to your application the following services:

- Allow users to enter any date, and then print the statistics for that day(consider all available years)..

---

<sup>1</sup> Even if it is better to define your `Date` class, you may also use `java.util.Date`.

- Allow users to enter any date in a month (month and day only), and then print the statistics for that day. These will include: The record high and low temps, the average high and low temp for that day and the average precipitation and snowfall amounts for that day (consider all available years).

- Allow the user to enter a month and then retrieve weather statistics for the given month. The statistics should include: The average high and low temp for the month, the range of high and low temps in that month, and the average rainfall and snow in that month.

In order to accomplish this, you will need to store all the data in a list. Use for this an array.

Use a main program which outputs for the following cases:

1. Statistics for December 1, 1956
2. Monthly statistics for the month of March
3. Statistics for April 1.

*Hint.* You could use overloaded methods.