

Project Five

The Temperature Statistics Problem

Write a program to produce the statistics for temperatures stored in a file. The centigrade temperatures in the file are the monthly averages for 12 months (January to December) in Cuba. Here is a sample displaying the required report:

```
Temperature Statistics in Fahrenheit
-----
Average temperature = 77.39
High Temperature   = August      82.04
Low Temperature    = January     71.96
```

Part I

main Driver - **TempDriver**

main should follow this sequence:

1. Declare an array of 12 doubles.
2. Declare a string with a value like this. It is the file path where the temp.dat file is stored.

```
String file = "e:\\temp.dat";
```

This **temp.dat** file is included with the assignment. Download and save the file on a storage device. You may need to change the drive designation in the string, depending on where you store the file. Windows 8/10 requires that a file be in a folder so this may be the path:
`c:\\folder\\temp.dat`

Mac Users (file path)

- Use the forward slash to separate directories instead of the back slash.
- Enter something like this if the file is on the desktop:
`/Users/Name/Desktop/temp.dat`
- The directory name and file name are case sensitive so you must use a capital letter where applicable.

*Name in the path will vary depending on your computer.

3. Call the **readFile** method that is part of the main driver file.

The **readFile** method receives the filename from main, reads the file into an array, and returns the array.

Use the *FileIO* file in *Chapter Seven* as an example of reading a file into an array. The *FileArrayExcept* file in the *Chapter Ten* materials is an example of displaying possible error messages when using the file.

Display an error message if the file is not found and add this statement after the error message:
System.exit(-1);

readFile returns an array of 12 values. These are the 12 monthly temperatures in the file.

Here is the structure of main before adding the **Temp** class code.

```
public class TempDriver
{//start of class

    main
    {//start of main
        declare array
        declare file string
        call readFile
        add a for loop to display the values in the array to make sure the array is correct.

    }//end of main

    readFile header
    {//start of readFile

    }//end of readFile

}//end of class
```

Review the contents of the *temp.dat* file before you start. You can open the file in the Eclipse editor by selecting *File/Open* and browse to find the data file.

Check the solutions to the Chapter Seven Lab – Problem 4 on the Discussion Board. These are examples of passing and returning arrays to a method. The second problem in the Chapter Ten Lab is practice with file processing and exceptions

Part II

Prepare an UML diagram for the **Temp** class to help you determine how to code the Temp class.

Class - **Temp**

Write a class to perform the statistics for any array of 12 temperatures.

The **Temp** Class – Be sure to *write and test one method at a time*.

Copy/paste the **months** array at the top of the class with the other private declarations.

```
private String [] months = {"January", "February", "March", "April", "May", "June", "July",
    "August", "September", "October", "November", "December"};
```

Temp has these *instance variables* in the *constructor*:

double **array**[] will be assigned the values passed to it using the **convertToFah** method described below.

```
double avg      = assign 0
double max     = assign 0
String maxMonth = assign null
double min      = assign 0
String minMonth = assign null
```

See the Chapter Seven Notes for an example of passing an array to a class constructor.

Temp contains these *methods*.

- **convertToFah** receives the array sent to the constructor and returns an array with the array values converted from Centigrade to Fahrenheit. Use a *for loop* to fill the array. The formula for converting Centigrade to Fahrenheit is
$$\text{Fahrenheit} = \text{Centigrade} * 9.0/5.0 + 32;$$
- **calcAvg** method calculates the average of the values in **array**, sets **avg** equal to the result, and returns the average, which is a double.
- **findMax** method finds the maximum value in **array** and its index in the array. **max** in the constructor is set to the maximum value and **maxMonth** is set to a month from the **months** array. The method returns the maximum value, which is a double.
Write the code assuming the values are not in ascending order. Do not sort the values.

- **findMin** method finds the minimum value in **array** and its index in the array. **min** in the constructor is set to the minimum value and **maxMonth** is set to a month from the **months** array. The method returns the minimum value, which is a double.
Write the code assuming the values are not in ascending order. Do not sort the values.
- **toString** method displays this output using the instance variables in the constructor:

```
Average temperature = 77.39
High Temperature   = August      82.04
Low  Temperature  = January     71.96
```

Make sure your output looks similar to this display.

Here is how you should approach the code in the Temp class.

- Declare the instance variables.
- Code the constructor.
- Code the *convertToFah* method.
- Add the appropriate code to main.
- Add a *for* loop in the constructor to display **array** after calling the *convertToFah* method to see that the values are correct.
- **Do not add any more code until the array contains the correct values.**
- Code the *calcAvg* method.
- Create the code in *toString* to display just the average until it works correctly.
- Add the appropriate code to main.
- Code the *findMax* method.
- Do not code *findMin* until *findMax* works.

All names in **bold** must be used as specified.

Project Points Check <i>Schedule</i> for due dates.	What to include
Project 5 - Part I (20 points)	<p>Submit <i>TempDriver (main)</i> with the <i>readFile</i> method.</p> <p><i>main</i> should only include the code needed to call the <i>readFile</i> method plus the <i>for</i> loop to show the contents of the array after reading the file. You will delete the <i>for</i> loop after the code works in the final version.</p>
Project 5 - Final Version (90 points)	Final Version of <i>tempDriver</i> and <i>Temp</i> classes

Purpose of this project

Develop Java program with these new features:

- o Arrays
- o File processing
- o Exceptions