

Project Report:

Critical Thinking 4 - Get Monthly Temperatures

Alejandro Ricciardi

Colorado State University Global

CSC320: Programming 1

Professor Herbert Pensado

May 19, 2024

Project Report:

Critical Thinking 4 - Get Monthly Temperatures

This documentation is part of the Critical Thinking 4 Assignment from CSC320: Programming 1 at Colorado State University Global. This Project Report is an overview of the program's functionality, pseudocode, and detailed testing scenarios including console output screenshots. The program is coded in Java JDK-21; and is named Critical Thinking 4 (Get Monthly Temperatures). The program is composed of a Main class and a MonthlyAvgTemps class.

The Assignment Direction

Get Monthly Temperatures

Develop a Java program that will store data in the form of monthly temperatures for a year. Store the month and temperature in two different arrays. Your program should prompt the user for the month to be viewed and display both the month and average temperature. If "year" is entered, the output for your program should provide the temperature for each month along with the yearly average as well as the highest and lowest monthly averages. *Use the looping and decision constructs in combination with the arrays to complete this assignment.*

Compile and submit your pseudocode, source code, and screenshots of the application executing the application, the results and GIT repository in a single document.

⚠ My notes:

- The program utilizes a string array to store the month names, the array's indexes correspond to the month names in yearly order. For instances:
The array index 0 corresponds to "January" → Number 1 (index + 1) is displayed to the user.
The array index 1 corresponds to "February" → Number 2 (index + 1) is displayed to the user.
....
The array index 10 corresponds to "November" → Number 11 (index + 1) is displayed to the user.
The array index 11 corresponds to "December" → Number 12 (index + 1) is displayed to the user.
- The month array can be found in the GradesStatCalculator class.
- The temperature array can be found in the Main class.
- **For the source code please see Main.java and GradesStatCalculator.java files.**

Git Repository

I use [GitHub](#) as my Distributed Version Control System (DVCS), the following is a link to my GitHub, [Omegapy](#).

My GitHub repository that is used to store this assignment is named [My-Academics-Portfolio](#) and the link to this specific assignment is:

<https://github.com/Omegapy/My-Academics-Portfolio/tree/main/Programming-1-CSC320/Critical-Thinking-4>

Pseudocode

```

Program: Grades Statistics Calculator
Author: Alejandro Ricciardi
Date: 05/19/2024
Description:
This program displays monthly average temperatures and allows the user to view individual monthly
temperatures or a yearly summary that includes the yearly average temperature as well as the
highest and lowest monthly averages.

Begin Program

-----
File: MonthlyAvgTemps.java
-----

Class MonthlyAvgTemps(array of temperatures)

--- Constructor ---
MonthlyAvgTemps(array of temperatures)
    Initialize highest to Integer.MIN_VALUE
    Initialize lowest to Integer.MAX_VALUE
    Initialize yearlyAvg to Integer.MIN_VALUE
    Initialize highestMonthIndex to 0
    Initialize lowestMonthIndex to 0
    Initialize months array with month names
    Assign temps array with provided temperatures

    Call computeStatsValuesIndexes()
End Method

--- Getters ---
Method getMonths()
    Return months array
End Method

Method getTemps()
    Return temps array
End Method

Method getHighestTemp()
    Return highest value

```

```

End Method

Method getLowestTemp()
    Return lowest value
End Method

Method getHighestMonthIndex()
    Return highestMonthIndex
End Method

Method getLowestMonthIndex()
    Return lowestMonthIndex
End Method

Method getYearlyAvg()
    Return yearlyAvg
End Method

--- Private Class Methods ---
Method computeStatsValuesIndexes()
    Initialize total to 0
    Iterate through temps array
        If temps[i] > highest
            Set highest to temps[i]
            Set highestMonthIndex to i
        End If
        If temps[i] < lowest
            Set lowest to temps[i]
            Set lowestMonthIndex to i
        End If
        Add temps[i] to total
    End Iteration Loop
    Set yearlyAvg to (Integer)Math.round(total / temps.length)
End Method
End Class

-----
File: Main.java
-----

Main Program
Class Main
    Initialize banner
    Initialize monthChoice
    Initialize quitProgram
    Initialize temperature array
    Initialize avgMonthlyTemps object with temperature array
    Initialize scanner

    Display banner

    Main Program while-loop(quitProgram is not equal to 'q')
        Display Menu choices

```

```

Prompt user to enter a month number or 'year' for yearly summary
If the user enters 'year'
    Call displayYearlySummary method
    Prompt the user to exit or continue the program
    If the user entered 'q'
        quitProgram = 'q'
    End If
Else If the user enters two-digit number
    Switch(two-digit number)
        The two-digit number is between 1 to 12
        Call displayMonthlyTemperature method
        with the corresponding month index equal to the entered number
        Prompt the user to exit or continue the program
        If the user entered 'q'
            quitProgram = 'q'
        End If
        The two-digit number is not between 1 to 12
        Display Invalid choice. Please enter a number between 1 and 12.
    End Switch
Else
    Display Invalid value entered, try again!
End Main Program while-loop if quitProgram = 'q'
Display Thank you for using the Get Monthly Temperatures program!
Exit Program
End Main Method

--- Main Class Method ---
displayMonthlyTemperature(MonthlyAvgTemps object, month and temperature corresponding index)
    Display the month and temperature from the corresponding index from the MonthlyAvgTemps
    object
End Method

displayYearlySummary(MonthlyAvgTemps object)
    months = get month names from the MonthlyAvgTemps object
    temps = get temperatures from the MonthlyAvgTemps object
    highestTempIndex = get the highest monthly average index from the MonthlyAvgTemps object
    lowestTempIndex = get the lowest monthly average index from the MonthlyAvgTemps object
    yearlyAvg = get yearly average from the MonthlyAvgTemps object

    Display all the months and corresponding temps using an iteration loop

    Display the highest average temperature month data from the months and temps arrays using
    The highestTempIndex
    Display the lowest average temperature month data from the months and temps arrays using
    The lowestTempIndex
    Display the yearlyAvg
End Method
End Class

End Program

```

The Main Class

The Main class is used to run the Get Monthly Temperatures program.

- It prompts the user to view and display both the month and average temperature.
- If "year" is entered, it displays the temperature for each month along with the yearly average as well as the highest and lowest monthly averages.
- Utilizes the MonthlyAvgTemps class.

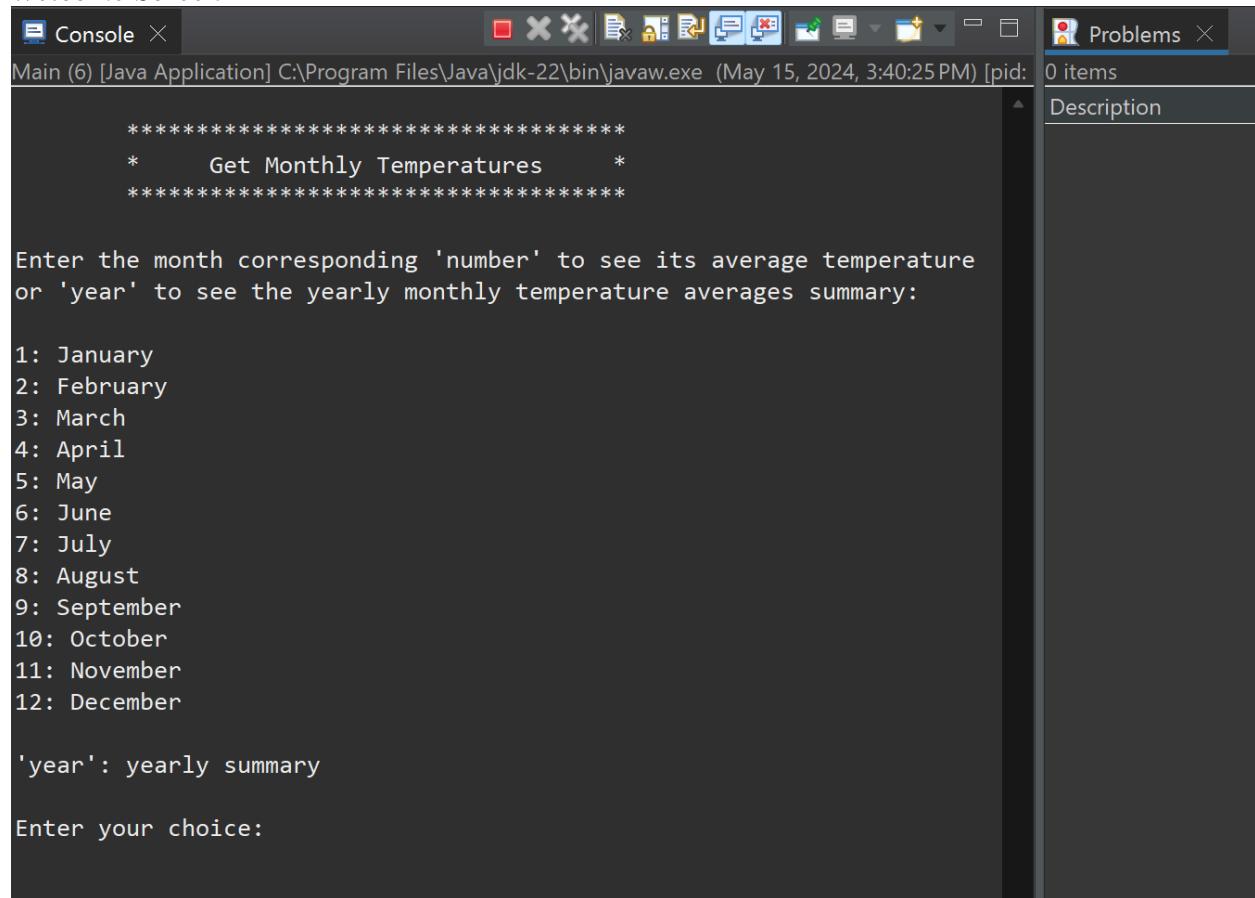
The GradesStatCalculator Class

The MonthlyAvgTemps class stores the average temperatures for each month.

- It computes the yearly average temperature as well as the highest and lowest monthly averages.
- Utilized by the Main class.

Screenshots: Program Functionality and Testing Scenarios

Figure 1
Welcome Screen



The screenshot shows the Eclipse IDE interface with two windows open:

- Console Window:** Displays the output of the Java application. The application starts with a title banner:

```
*****
*      Get Monthly Temperatures      *
*****
```

- It then prompts the user for input:

```
Enter the month corresponding 'number' to see its average temperature
or 'year' to see the yearly monthly temperature averages summary:
```

- A list of months is provided with numerical keys:

```
1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December
```

- The user is then prompted to enter their choice:

```
'year': yearly summary

Enter your choice:
```


- Problems Window:** Shows a list of items with 0 items present.

Note: Eclipse Console output from the program and Problem window showing no items.

Figure 2
Validation Tests Menu

```

Console X Main (6) [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (May 15, 2024, 3:40:25 PM) [pid: 28796]
*****
* Get Monthly Temperatures *
*****


Enter the month corresponding 'number' to see its average temperature
or 'year' to see the yearly monthly temperature averages summary:

1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December

'year': yearly summary

Enter your choice: Not a Number Not year
--- Invalid value entered, try again!

Enter the month corresponding 'number' to see its average temperature
or 'year' to see the yearly monthly temperature averages summary:

1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December

'year': yearly summary

Enter your choice: 0
--- Invalid choice. Please enter a number between 1 and 12.

Enter the month corresponding 'number' to see its average temperature
or 'year' to see the yearly monthly temperature averages summary:

1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December

'year': yearly summary

Enter your choice: 13
--- Invalid choice. Please enter a number between 1 and 12.

Enter the month corresponding 'number' to see its average temperature
or 'year' to see the yearly monthly temperature averages summary:

1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December

'year': yearly summary

Enter your choice:
--- Invalid value entered, try again!

Enter the month corresponding 'number' to see its average temperature
or 'year' to see the yearly monthly temperature averages summary:

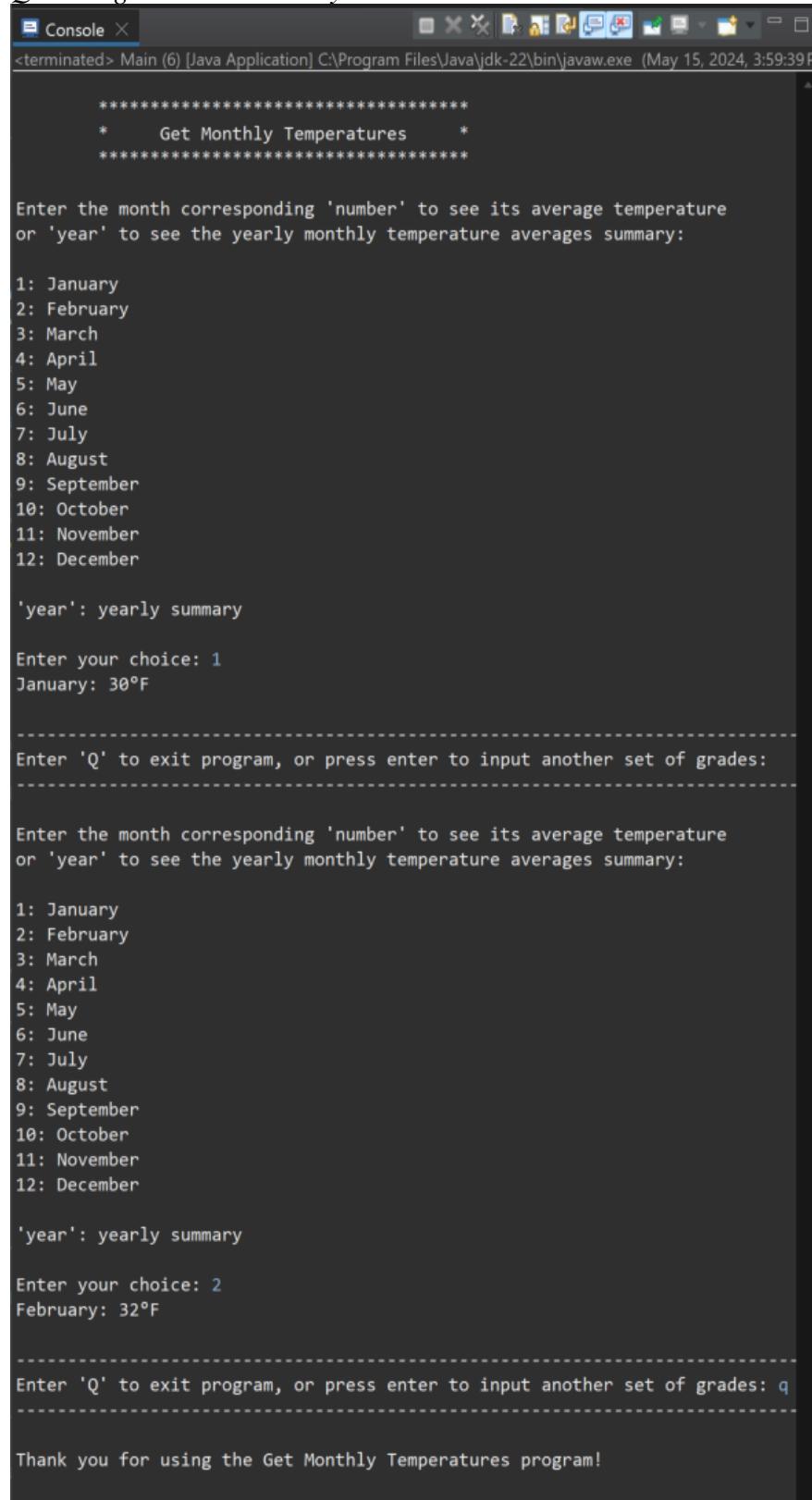
1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December

'year': yearly summary

Enter your choice:

```

Figure 3
Quit Program Functionality



The screenshot shows a Java application running in a console window. The title bar reads "Console" and the path "Main (6) [Java Application] C:\Program Files\Java\dk-22\bin\javaw.exe (May 15, 2024, 3:59:39 PM)". The window displays the following text:

```
*****  
*      Get Monthly Temperatures      *  
*****  
  
Enter the month corresponding 'number' to see its average temperature  
or 'year' to see the yearly monthly temperature averages summary:  
  
1: January  
2: February  
3: March  
4: April  
5: May  
6: June  
7: July  
8: August  
9: September  
10: October  
11: November  
12: December  
  
'year': yearly summary  
  
Enter your choice: 1  
January: 30°F  
  
-----  
Enter 'Q' to exit program, or press enter to input another set of grades:  
-----  
  
Enter the month corresponding 'number' to see its average temperature  
or 'year' to see the yearly monthly temperature averages summary:  
  
1: January  
2: February  
3: March  
4: April  
5: May  
6: June  
7: July  
8: August  
9: September  
10: October  
11: November  
12: December  
  
'year': yearly summary  
  
Enter your choice: 2  
February: 32°F  
  
-----  
Enter 'Q' to exit program, or press enter to input another set of grades: q  
-----  
  
Thank you for using the Get Monthly Temperatures program!
```

Figure 4
Menu Functionality – Month Averages

```

Console <terminated> Main [6] [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (May 15, 2024, 4:05:04)

*****  

* Get Monthly Temperatures *  

*****  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 1  

January: 30°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 2  

February: 32°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 3  

March: 45°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 4  

April: 55°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 5  

May: 65°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 6  

June: 75°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 7  

July: 80°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 8  

August: 78°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 9  

September: 70°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 10  

October: 60°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 11  

November: 50°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c  

-----  

Enter the month corresponding 'number' to see its average temperature  

or 'year' to see the yearly monthly temperature averages summary:  

1: January  

2: February  

3: March  

4: April  

5: May  

6: June  

7: July  

8: August  

9: September  

10: October  

11: November  

12: December  

'year': yearly summary  

Enter your choice: 12  

December: 35°F  

-----  

Enter 'Q' to exit program, or press enter to input another set of grades: c

```

*Figure 5**Menu Functionality – Yearly Average*

Enter the month corresponding 'number' to see its average temperature or 'year' to see the yearly monthly temperature averages summary:

```
1: January
2: February
3: March
4: April
5: May
6: June
7: July
8: August
9: September
10: October
11: November
12: December
```

```
'year': yearly summary
```

```
Enter your choice: year
```

Monthly Average Temperatures:

January: 30°F

February: 32°F

March: 45°F

April: 55°F

May: 65°F

June: 75°F

July: 80°F

August: 78°F

September: 70°F

October: 60°F

November: 50°F

December: 35°F

Yearly Average Temperature: 56°F

Highest Monthly Average: July with 80°F

Lowest Monthly Average: January with 30°F

```
-----  
Enter 'Q' to exit program, or press enter to input another set of grades: q  
-----
```

Thank you for using the Get Monthly Temperatures program!

As shown in Figure 1 through Figure 5 the tests run without any issues, displaying the correct output as expected.