SEP105: Introduction to Programming for Engineers

Project 3

T2 2023

Due Date: See the unit site

Written by: Benjamin Champion

Project 3 – Description

The Turing Moore Engineering company has been tasked with building composite halls for new boats. Different types of resin cure best in different ambient temperature conditions. You have been tasked with designing a program to help determine the best times of year to manufacture the different halls for the boats, based off the temperature ranges for those months.

As a minimum, the program must display a Welcome message as the first thing displayed in the program with the following information, each on a separate line:

- The message "Turing Moore Engineering Temperature Data Analyser 1.0"
- Your name (as shown on the unit site)
- Your Student ID
- Date the assignment is due in the format: dd/mm/yyyy
- The level of mark you are aiming to achieve (eg Desired Level: HD)

An example of what this **may** look like is shown below:

Turing Moore Engineering Temperature Data Analyser 1.0
Name: Benjamin Champion
Student ID: 123456789
Desired Level: C

The following points apply to this document:

- This assessment is worth 45% of the total grade.
- The program must be written entirely by you!
- The program must be written in C++ targeted at the Windows Operating System. Please use Visual Studio Community Edition to write and compile your code on a Windows PC
- Only functions, types, etc that was covered in class between **weeks 1-11** can be used in this assessment. Anything else being used may be considered for academic misconduct.
- There will be an oral defence of the code you produce for this assessment.

The following information describes the requirement for the different levels in this assignment. Before completing tasks from a higher level all tasks from a lower level must first be completed.

Pass (P) Level (50 – 59):

To achieve a pass for the unit the software must be able to display the following behaviour:

- Prompt the user with an appropriately formatted message about how to use the software.
- Ask the user if they want to view Bendigo Airport data, Avalon Airport Data or both. The data for each file can be found in the associated .csv files downloaded from the unit site.
- Display the total number of rows and columns in the file.
- From the selected data set, find the hottest day of the year and display the temperature, date and year.
- From the selected data set, find the coldest day of the year and display the temperature, date and year.
- From the selected data set, find the mean temperature for the year
- From the selected data set, find the median temperature for the year

- All information written to the terminal, and inputted by the user, should also be saved to a file called “log.txt”
- Ask the user if they would like to continue by entering the character ‘c’ or exit by entering the character ‘e’. If continue is selected, the program should repeat from just after displaying the welcome message. If exit is selected, the program should end, and an appropriate message should be displayed.

Credit (C) Level (60 – 69):

- All functionality of the Pass level
- Find the mode of the data set to the closest degree (eg 18.3 becomes 18. You do not need to round the values).
- From the selected data set, find the month with the coldest average temperature for the year and display the month, along with temperature value, to the user.
- From the selected data set, find the month with the hottest average temperature for the year and display the month, along with temperature value, to the user.
- From the selected data set, find the month with the most consistent data set for the year. The most consistent month will be the month with the smallest variance in the temperature values. Display the month, the average temperature and the variance to the user. See Required Theory for information on how to calculate the variance.
- In a new folder called Results, make a new .csv file called results.csv. The columns should have the following headings, with the associated data for each month on the associated row: “Year, Month, Hottest Temperature, Coldest Temperature, Average Temperature, Median Temperature”.

Distinction (D) Level (70 – 79):

- All functionality of the Credit level
- Show the percentage of days that fall within one standard deviation of the mean of the data set. Display the percentage of days and the value for the standard deviation to the user. See Required Theory for information on how to calculate the Standard Deviation.
- The users should be giving the option of choosing the files with data for all recorded years or the file for a single year. A message should be displayed to the screen telling the user the start date and end date of the file
    - Hint: Some information may be missing in the file. This will be indicated by a blank cell if the file is opened in a spreadsheet program such as excel. If the file is opened in a notepad program, such as Notepad++, it will be indicated by two commas directly next to each other (,,). Your program will need to be able to determine if there is missing information. If the information is missing, then that date should be excluded. Please note, this will effect some of your calculations, such as calculating the mean, as the total number of samples will be less!
- In the results folder, instead of making one file called “results.csv”, a new file for each year in the opened file should be made containing the information relative to that year. The name of each file should be the name of the year. For example, for data pertaining to 2001, the file would be called “2001.csv”. If a month in a year does not have any data (for example Avalon did not start recording data until April of 1995), the data field should be left blank for that month in the result file. A row for that month should still be present.

High Distinction (HD) Level (80 – 100):

- All functionality of the Distinction level
- The user should be shown all available years in the file. The user then should be able to select a specific year, or select all years, for information to be displayed to the user. For example, if 2001 is chosen, the hottest, coldest, etc day in 2001 would be displayed.
- Alternatively, the user should be able to select a month and the following information should be displayed to the user:
    - The date of the hottest day for that month on record
    - The date of the coldest day for that month on record
    - The year with the most consistent temperature (lowest variance)
- The user should be given another option to open any file. The name of this file should be entered by typing in the file name. It is assumed that this file is in the same format as the other files.
- If there is an error in the file, for example a character where a digit was expected, the program should tell the user what row and column an error first appeared on and stop processing the file.

Appendix

Background

The Turing Moore Engineering company has been tasked with designing new components out of composites for boats. Different parts will use different composite materials, which all have optimal curing temperatures. Some cure best under hot temperatures, some cure best under cold temperatures, some cure better with consistent temperatures and some cure best under highly varying temperatures. You have been tasked with designing a program to help determine the best times of year to manufacture the different components for the boats, based off the temperature ranges for those months.

As it is not possible to predict the weather conditions with 100% accuracy long term in the future, you have decided to look towards historical data to base your results off. You have found that public temperature data is available close to all of your manufacturing sites on the Bureau Of Meteorology (BOM) website: http://www.bom.gov.au/climate/data/. The two sites you are going to start manufacturing in are located next to Avalon and Bendigo airports in Victoria. Using this data, you should be able to get a best guess at the right times of the year to manufacture components using the different types of resin to get the best result.

Prior knowledge

At a **minimum** to pass this assessment you will need to have skills in the following areas. If you feel like you are not as strong in these areas as you would like to be, I recommend you go and brush up on the class content. Please note, depending on how you write your program, and the desired level you are aiming for, you may not use everything listed. That is OK as there are always multiple ways to write a program with the same functionality.

- Flow Diagrams
- Initialising and manipulating variables (int, float, double, etc)
- Reading and Writing from the console

- Mathematical Operations
- Conditionals (&&, ||, etc)
- Decisions (if, switch)
- Debugging
- Looping (for, while, do while)
- Arrays
- Structures
- Vectors
- Pointers
- Functions
- Classes
- File I O
- Custom Libraries

What to submit

A .zip folder containing the following items:

- A flow chart as a PDF – it is highly recommended that you make your flowchart **BEFORE** writing any software.
- All fully commented code files that pertain to the project (eg the main.cpp file). These files must be the same files that generated the submitted executable file.
- An .exe of the final code – please note this executable will be tested on a Windows PC. It is up to you to ensure that the software that you write will execute correctly on a Windows PC. If you use a Windows PC and the methods that are described in class, the code will run and execute on a Windows PC. If you use some other operating system (eg MAC, Linux, ChromeOS) there is no guarantee that the software you submit will run, or run correctly. Consider this your only warning, any software that does not run for this reason will generate 0 marks.
- Any files that you have been asked to generate or read into the program (for example .csv or .txt files).
- The entire project folder containing the software. This is independent to the other documents you are required to submit but should be in the same .zip folder.

Hints

- It is recommended that you develop the program in stages, do not shoot for a High Distinction straight away. First make a flowchart and program that will achieve a pass. Save this, make a new program, copy in the previous solution and modify the flow chart and program to reach a credit, etc. Make sure you have all of the functionality working of the previous level **BEFORE** moving onto the next level. You only need to submit the most recent working version of the program you do not need to submit earlier versions of the program.
- Make your flowchart(s) **BEFORE** you write any software. This will help you reduce the amount of code you need to write and help you significantly in debugging your code!
- While it is tempting to use AI such as Chat GPT to help write your code, in this unit it will be more of a hindrance than a help! As these AI systems do not know what we have covered in class, they will most likely give you code that will not be accepted. Often the code written by these systems is not 100% correct and therefore you will need to manually fix it. You can only do this if you understand the code in the first place (something you will gain in this unit). Finally, you may be asked about your code, how it works, etc. If you have not written the code,

you will not be able to answer these questions. This could result in 0 for the assessment, and possibly academic misconduct.

- Do not modify the temperature files given for testing your program. I recommend that you redownload the files from the unit site again and test your program to make sure. We will be using a fresh copy of the temperature files uploaded to the unit site to test the program. We may also test your program using data from a different site.
- If you wish to test your program out on data from other locations around Australia, more temperature information can be downloaded from the Bureau of Meteorology (BOM) directly: http://www.bom.gov.au/climate/data/. Please note that the raw files downloaded from the BOM may be missing information for some days. For the 2022 data for both Avalon and Bendigo uploaded to the unit site, the teaching team have modified the file to remove any blank days. The files containing all year's information have been downloaded without any modification. Therefore, only a program completed to the D level will be able to accurately process the raw files directly from the BOM.
- View the information in Required Theory for information about some of the calculations required for the project.

Required Theory

Mean

The mean of a data set can also be called the average of a data set and is represented by:

$$mean = \frac{\sum_0^n x_n}{N}$$

Where x_n is the n^th value in the set, and N is the number of values in the set. For example, if we had the numbers:

1, 8, 5, 3, 26, 4, 7

The mean would be:

$$mean = \frac{\sum_0^n x_n}{N} = \frac{\sum_0^6 x_n}{N}$$

$$mean = \frac{(1+8+5+3+26+4+7)}{7}$$

$$mean = 7.86$$

Median

The median is the middle number of a set of data. The median value is sometimes used over the mean as it is not affected by outliers in the data set as much. To find the median value, simply order the data set from largest to smallest, and pick the middle number. For example, if we use the same numbers:

1, 8, 5, 3, 26, 4, 7

First, you order the numbers:

1, 3, 4, 5, 7, 8, 26

And then pick the middle number (so there is the same number of numbers on both the left and the right):

$median = 5$

If the total number of elements is even, then there is no middle number as the set can be broken into two even groups. In this case, the two middle numbers are chosen and averaged to find the median. For example, if we had this set of numbers:

5, 4, 6, 3, 8, 7

First, order the numbers:

3, 4, 5, 6, 7, 8

Find the middle numbers. In this case, we have two, so they are:

5 and 6

Now, find the average between the numbers:

$median = \frac{5+6}{2} = 5.5$

Mode

The mode of a data set is simply the number(s) that appear the most in the data set. It is possible to have multiple mode's in the data set, or to have none at all (assuming all numbers only appear once or an equal number of times).

For example, given the following data set, find the mode(s):

1, 8, 1, 7, 3, 6, 2, 4, 1, 9

In this case, 1 appears 3 times. This is the most times, so it is the mode of the data set.

Lets look at another example. Consider this data set:

2, 8, 6, 4, 2, 5, 9, 7, 2, 4, 6, 4

In this case, 2 appears three times and so does 4. Therefore, they are both the mode of the data set.

Variance

The variance can be used to compare different data sets to each other to determine how close on average the values in the data set are to each other. This should mean that a data set with values that are generally closer to each other should have a lower variance than a data set with values that are on average a longer way apart from each other.

Calculating the variance is relatively straightforward. We can use the following equation:

$$\sigma^2 = \sum_{0}^{n} |x_n - \mu|^2$$

Where σ^2 is the variance, x_n is the nth point in the set and μ is the mean of the set.

For example, let's consider the following data set:

1, 3, 5, 7, 14

First, calculate the mean of the data set:

$$mean\ (\mu) = \frac{\sum_0^n x_n}{N} = \frac{\sum_0^4 x_n}{N}$$
$$mean\ (\mu) = \frac{(1+3+5+7+14)}{5}$$
$$mean\ (\mu) = 6$$

Next, can take each value in the set away from the mean and square it:

Value (x)	$\|x - \mu\|^2$
1	25
3	9
5	1
7	1
14	64

Finally, sum them all together to get the variance:

$$\sigma^2 = \sum_0^n |x_n - \mu|^2$$

$$\sigma^2 = 25 + 9 + 1 + 1 + 64$$

$$\sigma^2 = 100$$

Standard Deviation

The standard deviation can be used to measure how clustered data is around the mean. The larger the data is spread out, the larger the standard deviation will be! To calculate the standard deviation, the following equation is used:

$$SD = \sqrt{\frac{\sigma^2}{N}}$$

Where x is the current value, σ^2 is the variance of the data set and N is the total number of elements in the data set.

For example, let’s consider the following data set:

8, 5, 8, 4, 3, 2, 6, 8, 9

First, we need to find the mean of the data set:

$$mean\ (\mu) = \frac{\sum_0^n x_n}{N} = \frac{\sum_0^8 x_n}{N}$$
$$mean\ (\mu) = \frac{(8+5+8+4+3+2+6+8+9)}{9}$$
$$mean\ (\mu) = 5.89$$

Next, we can find the variance of the data set:

Value (x)	$\|x - \mu\|^2$
8	4.45

5	0.79
8	4.46
4	3.57
3	8.35
2	15.12
6	0.01
8	4.46
9	9.68

$\sigma^2 = \sum_0^n |x_n - \mu|^2 = 50.89$

And finally, we can calculate the standard deviation:

$$SD = \sqrt{\frac{\sigma^2}{N}}$$

$$SD = \sqrt{\frac{50.89}{9}}$$

$$SD = 2.38$$

To find a percentage of how many values fall within one standard deviation of the mean simply order the data set:

2, 3, 4, 5, 6, 8, 8, 8, 9

Calculate a value either side of the mean:

Lower limit: $\mu - SD = 5.89 - 2.38 = 3.51$

Upper limit: $\mu + SD = 5.89 + 2.38 = 8.27$

Then count the values that fall between the Upper and Lower limit. In this case, we have 3 values (4, 5 and 6) that fall within one standard deviation of the mean, or 33% of values. Generally, this analysis is performed on data sets much larger than this to show any real benefit.

Rubric

All Assignments will be marked off the following Rubric. Please note:

- No marks will be awarded for a higher grade for the software until **all of the functionality** of the lower grade is awarded. For example, if you only have partial parts of a Credit working, and some parts of a Distinction working, the highest mark you will be able to achieve is in the Credit range
- The flow chart must represent the code that was submitted. Do the flowchart before writing any software! If the flow chart includes some functionality that was not included in the final solution, this is OK. Code including functionality not included in the flow chart is not accepted and you may be heavily penalised.
- The submitted code must be the same as the executable file that was submitted. If this is not the case, ie the functionality of the code does not match what is observed in the executable, it will be assumed that you have not written the code and you may be subject to academic misconduct.

See next page for the rubric

	HD	D	C	P	F
Flow Chart Presentation	The flow chart is laid out exceptionally well. Is very easy to read, understand and follow all aspects of the flow chart. The correct symbols are used for all or most sections of the flow chart. All arrows are labelled where appropriate.	The flow chart is laid out exceptionally well. Is very easy to read understand and follow most components of the flow chart. The correct symbols are used for all or most sections of the flow chart. All arrows are labelled where appropriate.	The flow chart is laid out in a way that makes sense. Is easy to read, understand and follow most aspects of the flow chart. The correct symbols are used for all or most sections of the flow chart. All arrows are labelled where appropriate.	The flow chart is laid out in a way that makes sense. The flow information in the flow chart can be read, makes sense and the flow can be followed. The correct symbols are used for all or most sections of the flow chart. All arrows are labelled where appropriate.	Either the flow chart is missing some information, is laid out in a way that is difficult to follow, cannot be read, consistently uses the wrong symbols or does not label all arrows when appropriate.
Flow Chart Function	The flow chart shows some or all components for a HD level program. The function of all components make sense. The flow chart either exceeds or matches the functionality of the program	The flow chart shows some or all components for a D level program. The function of all components make sense. The flow chart either exceeds or matches the functionality of the program	The flow chart shows some or all components for a C level program. The function of all components make sense. The flow chart either exceeds or matches the functionality of the program	The flow chart shows some or all components for a P level program. The function of all components make sense. The flow chart either exceeds or matches the functionality of the program	The flow chart either is not present, does not show enough information or functionality, or does not resemble the functionality of the code.
Comments	All aspects of the code are fully commented. This includes all function blocks, header blocks, etc where appropriate. All lines of the code have comments describing their purpose and is not simply a redefinition of the code.	All aspects of the code are fully commented. This includes all function blocks, header blocks, etc where appropriate. Most lines of the code have comments describing their purpose and is not simply a redefinition of the code.	Most aspects of the code are commented. This includes all function blocks, header blocks, etc where appropriate. Most lines of the code have comments describing their purpose and is not simply a redefinition of the code.	Most aspects of the code are commented. This includes all function blocks, header blocks, etc where appropriate. Either little inline comments have been provided, or the comments that are provided are simply restating the code.	The code either has little to no comments, or is missing some required information such as header blocks or function blocks.
Code Structure	The code follows all aspects of the style guide. The code is written in a way that is very easy to read and follow.	The code follows all aspects of the style guide. The code is written in a way that is easy to read and follow	The code follows all aspects of the style guide. The way the code is written and can be followed, but could be improved	The code follows most aspects of the style guide. The way the code is written can be followed, but could be improved	The code does not follow the style guide. For example, the variables may be using the wrong naming convention, indentation may not be correct, etc. The code may also be written in a way that is difficult to read and/or follow
Program Functionality	The program demonstrates some or all functionality for a HD level as described in the assignment description. All functionality for the previous levels is present.	The program demonstrates some or all functionality for a D level as described in the assignment description. All functionality for the previous levels is present.	The program demonstrates some or all functionality for a C level as described in the assignment description. All functionality for the previous levels is present.	The program demonstrates most of the functionality for a P level as described in the assignment description.	The program does not show all or most of the functionality for a P level as described by the assignment.
Programming Techniques used	The program uses all techniques covered in class at least once. These include (but not exclusive to): Libraries Classes Functions Pointers File IO	The program uses all techniques covered in class at least once. These include (but not exclusive to): Libraries Classes Functions Pointers File IO	The program uses most techniques covered in class at least once. These include (but not exclusive to): Libraries Classes Functions Pointers	The program uses some techniques covered in class at least once. These include (but not exclusive to): Libraries Classes Functions Pointers	The program uses some techniques covered in class, but not very often or in ways that don't make sense.
Defence of the project	The defence is a hurdle, and you have passed it. All questions asked by the assessors about the submitted code and/or flowchart could be answered to an acceptable level by the student				Questions could not be answered to an acceptable level