

**Schulich School of Engineering
University of Calgary**

ENGG 233 –Fall 2020

Term Project: Canada Weather Analysis

Due Dates	
Pre-project and Milestone 1 Demo	<p>Submit Electronically on D2L and Demo:</p> <ul style="list-style-type: none">• Wednesday lab (Section B2):<ul style="list-style-type: none">○ <u>Due on Wednesday November 25th, 2:59 PM</u>• Thursday Lab (Section B4):<ul style="list-style-type: none">○ <u>Due on Thursday November 26th, 11:59 AM</u>• Friday Lab (Section B1):<ul style="list-style-type: none">○ <u>Due on Friday November 27th, 9:29 AM</u>• Friday Lab (Section B3):<ul style="list-style-type: none">○ <u>Due on Friday November 27th, 1:59 PM</u>
Milestone 2 Demo	<p>Submit Electronically on D2L and Demo:</p> <ul style="list-style-type: none">• Wednesday lab (Section B2):<ul style="list-style-type: none">○ <u>Due on Wednesday December 2nd, 2:59 PM</u>• Thursday Lab (Section B4):<ul style="list-style-type: none">○ <u>Due on Thursday December 3rd, 11:59 AM</u>• Friday Lab (Section B1):<ul style="list-style-type: none">○ <u>Due on Friday December 4th, 9:29 AM</u>• Friday Lab (Section B3):<ul style="list-style-type: none">○ <u>Due on Friday December 4th, 1:59 P</u>

Project Objectives

1. To gain experience with the design and development of a relatively larger software project.
2. To gain practical experience in using programming constructs learned in ENGG233 such as classes, arrays, functions, etc.
3. To gain experience with the iterative development process.
4. To develop teamwork and presentation skills as a programmer.

Submissions and Group Work

Students have the option of working on the project in groups of two or working alone. Groups of larger than 2 people will not be allowed.

If students choose to work in groups, **they must choose partners from within their TA lab groups.**

Instructions for Groups

Students who work in groups must follow the following guidelines and regulations:

1. The workload **must be divided equally** between the two students. During the demos, the TAs will specifically ask about what each student has done in order to ensure both students have contributed equally to the project.
2. Both students must be present at each demo. If a student is absent, that student will not be awarded the grades for the demo.
3. Only one of you needs to submit each milestone to D2L. However, please make sure **both of your names are included** in your submission.

Important Note:

You can work with other groups and students, however, be very careful not to simply use the code from other students. Plagiarism is a serious academic offense.

Pre-project Exercise:

Data manipulation and Data Analysis by using NumPy

You will get familiar with Numpy library, which is used for numerical computation in python.

NumPy is a library for the Python programming language that adds support for large, multi-dimensional arrays and matrices. A reference for using NumPy is available through the following link: <https://numpy.org/doc/stable/reference/index.html>

To install the packages, please run the following commands in your terminal

Installing matplotlib (which includes NumPy)

```
# macOS

python3 -m pip install matplotlib

# Windows (may require elevation)

python -m pip install matplotlib

# Linux (Debian)

apt-get install python3-tk

python3 -m pip install matplotlib
```

In the project, you will learn how to use the numpy package which is a very common numerical package in python. The project is divided into two milestones, in the first milestone you will be asked to develop basic classes and methods for weather data analysis. In the second milestone, the focus of the project would be on data analysis and data visualization where numpy package and matplotlib can be used, respectively.

Numerical Analysis by numpy

Ex1:

Initialize a random array with length of 10 and calculate the following parameters of the list:

- minimum element of the array
- maximum element of the array
- average value of array
- median of array
- standard deviation of array (in case you don't know what's standard deviation, here is an excellent link explaining it. You will learn more in ENGG319. <https://numpy.org/doc/stable/reference/generated/numpy.std.html>)

For example, here is an example of randomly generated array and its related features:

```
[82 27 71  0 27 54 79 25 95 11]
min:  0
max:  95
average:  47.1
median:  40.5
std:  31.57039752679716
```

Reading CSV file by numpy and visualization

Ex2:

Imagine you are hired as a data analyst in a software company and they provide you a csv file which includes the information of calgary weather in 2000 monthly. By using the following source code read the csv file. Then, follow the following steps to visualize the data temperature according to the year.

What to do:

Step1)

Write a main function that can read the weather.csv file and store the information in a numpy **data** variable. Print **data** to make sure that you have read the file correctly.

The source code for reading data from csv file:

```
import numpy as np

def read_weather():
    # 1- pre exercise sample reading csv file by using numpy
    file_name = "weather.csv"

    data = np.loadtxt(file_name , delimiter=',', skiprows=1,
                      usecols=(0,1,2,3,4), dtype=np.float)
    return data
```

Fig. 1

Step2)

make two lists: **x** and **y**. Append months information to **x** list and for **y** list append the minimum temperature of corresponding month. Then, call the **drawChart** function and pass **x** and **y** to the function to draw the graph. **Note:** Make sure that you are using the correct labeling and title for your graph.

```
import matplotlib.pyplot as pyplot

def drawChart(x,y):

    fig = pyplot.figure()
    pyplot.title('Temperatures in Calgary between Jan-Dec in 2000')
    pyplot.ylabel('Min Temperature (F)')
    pyplot.xlabel('Month of Year')

    pyplot.plot(x, y, marker='o')
    pyplot.show()
```

Fig. 2

Fig. 3 is the expected graph.

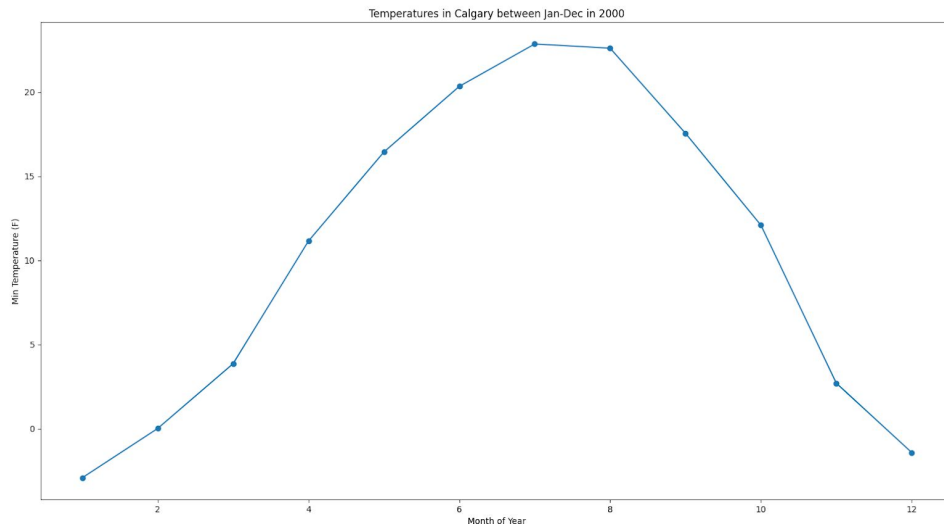


Fig. 3

Project:

Instruction:

You have been hired by the city to visualize some simple weather data. The data provides the average daily minimum and maximum temperature for each month for about 30 years. Your job is to analyze this data and visualize some important features from the given data.

The project is divided into two milestones

- Back-end development (Object Oriented Programming)
- Data Analysis (feature extraction and data visualization)

You need to visualize the data using the following charts:

- Bar chart (Guide: <https://pythonspot.com/matplotlib-bar-chart/>)
- Line chart (Guide: Fig. 2)

The Data:

The data used for this project has been downloaded from the following source:

<https://calgary.weatherstats.ca/charts/precipitation-monthly.html>

and has been simplified for your project. Please download the “.csv” file called CalgaryWeather.csv from D2L to use for this project.

The minimum requirements of the project are divided into visualization and data analysis:

Requirements (Checklist)

- **milestone 1: (35 Marks)**
 - Get the minimum temperature of 1990-2019
 - Get the maximum temperature of 1990-2019
 - Get the minimum temperature of 1990-2019 annually
 - Get the maximum temperature of 1990-2019 annually
 - Get the average snowfall between 1990-2019 annually
- **milestone 2: (45 Marks)**
 - Get the average temperature between 1990-2019 annually
 - draw the line chart for minimum temperature of 1990-2019 (x-axis is year, y-axis is minimum temperature each year)
 - draw the line chart for maximum temperature of 1990-2019 (x-axis is year, y-axis is maximum temperature each year)
 - draw the bar chart for average snowfall of 1990-2019 (x-axis is year, y-axis is average snowfall in each year)
 - draw the bar chart for average temperature of 1990-2019 (x-axis is year, y-axis is average temperature in each year)
- **Report: (20 Marks)**

Program Modules and Classes:

Your program must have several classes that will be organized as separate files. Here is the list of some of the modules or classes that your program should have with the minimum requirements below - note that you will need to add member functions and possibly extra member variables to the classes listed below - :

- **Class FileIO**

The class has the following member variables

- filePath to store the path of the selected sensor data file
- dataTable of type numpy to load the data in the csv input file

- **Class Date**

Member variables :

- Month
- Year

- **Class TemperatureData**

Member variables :

- A Date object
- minTemperature
- maxTemperature
- snowFall

- **Class Chart**

These classes handle the plotting of each type of graph. They would have member functions such as

- drawLineChart(x,y,title,xlabel,ylabel)
- drawBarChart(x,y,title,xlabel,ylabel)

- **Class WeatherAnalyzer**

Member variables :

- list of temperatureData objects

The methods inside weatherAnalyzer:

- getMinTemp
 - return the minimum temperature of the given data
- getMinTempAnnually
 - return the minimum temperature of the given data according to each year
- getMaxTemp
 - return the maximum temperature of the given data
- getMaxTempAnnually
 - return the maximum temperature of the given data according to each year
- getAvgSnowFallAnnually
 - return a 2D list of each year average snowfall
- getAvgTempAnnually

- return the average temperature of each year (note: average temperature of each month is calculated based on $(\text{min_temp} + \text{max_temp})/2$)

You also need to develop a menu bar as below, where users can enter each section and get the result in the console.

```
1- Get Minimum Temperature of 1990-2019
2- Get Maximum Temperature of 1990-2019
3- Get Minimum Temperature of 1990-2019 Annually
4- Get Maximum Temperature of 1990-2019 Annually
5- Get Average Snowfall between 1990-2019 Annually
6- Get Average Temperature between 1990-2019 Annually
7- LineChart Minimum Temperature of 1990-2019 Annually
8- LineChart Maximum Temperature of 1990-2019 Annually
9- BarChart Average Snowfall between 1990-2019 Annually
10- BarChart Average Temperature between 1990-2019 Annually
```

Project Requirements in Iteration

software projects can be very complex, therefore the process used to manage this complexity is very important. One effective process widely used in software development is referred to as iterative development.

Simply put, iterative development is the processes of breaking down a larger project into smaller steps (i.e. iterations or milestones). To ensure you are on track for the project, and to make sure you will get marks for all your efforts, not just for the completed project, this project is divided into 2 different milestones. For each milestone, you should fully test your code and ensure you satisfy all the requirements for that milestone correctly. This is very important as you don't want to keep building on a faulty project.

Milestone1:

In this milestone, you are expected to implement the infrastructure of the program which has FileIO (reading from the csv file), Date, TemperatureData, WeatherAnalyzer classes and also main function. In the first milestone, your program should implement first five functionality of the menu:

```
"1- Get Minimum Temperature of 1990-2019",
"2- Get Maximum Temperature of 1990-2019",
"3- Get Minimum Temperature of 1990-2019 Annually",
"4- Get Maximum Temperature of 1990-2019 Annually",
```

"5- Get Average Snowfall between 1990-2019 Annually",

For example, if the user selects option 2, the program should print the maximum temperature between 1990-2019.

What to submit for Milestone 1:

Compress all .py files in a zip file and upload the zip file. The compressed file must include the following files, where each contains the implementation of the corresponding class:

- Chart.py
- Date.py
- FileIO.py
- main.py
- TemperatureData.py
- WeatherAnalyzer.py

Milestone2:

In this milestone you are expected to implement the rest of the project including:

"6- Get Average Temperature between 1990-2019 Annually",
"7- LineChart Minimum Temperature of 1990-2019 Annually",
"8- LineChart Maximum Temperature of 1990-2019 Annually",
"9- BarChart Average Snowfall between 1990-2019 Annually",
"10- BarChart Average Temperature between 1990-2019 Annually"

If a user chooses option 7, 8, 9, or 10, the program should visualize the results. For example, For option 7 and 8 the results should be presented in a line chart. For example, if the user selects option 7, the output would be like Fig. 4.

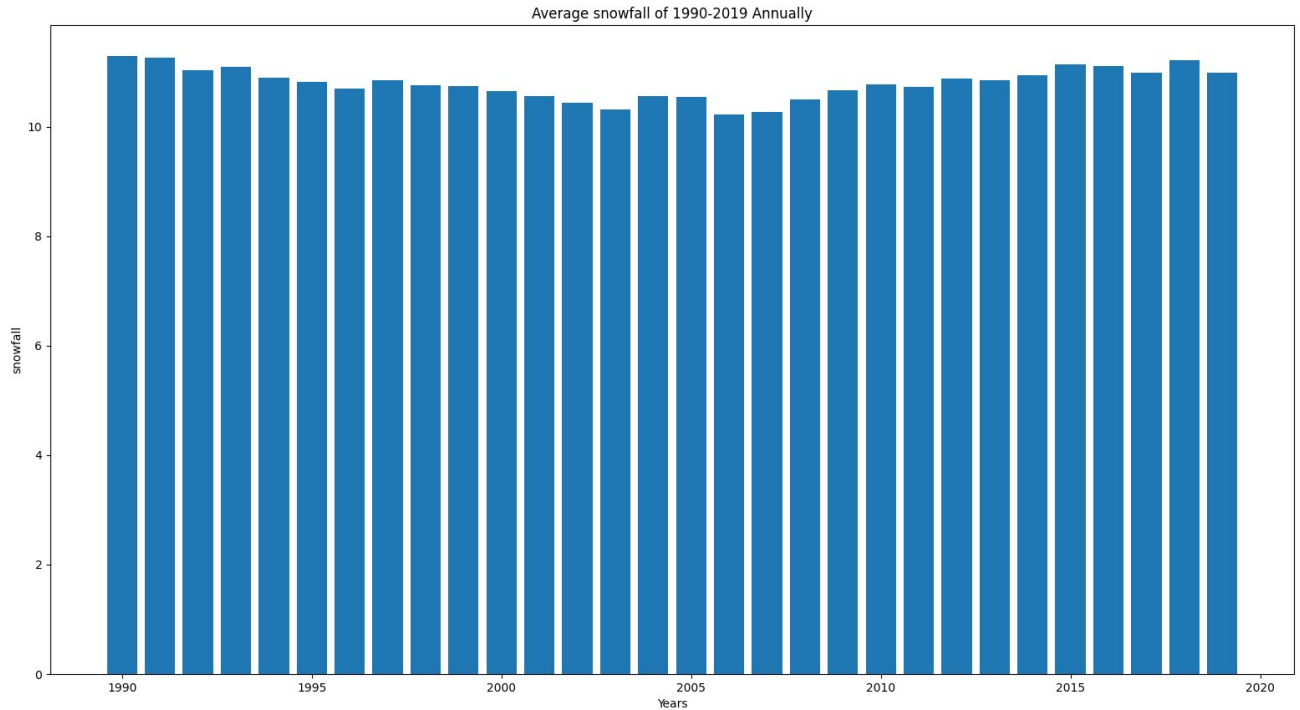


Fig. 4

What to submit for Milestone 2:

Compress all .py files in a zip file and upload the zip file. **Also, prepare a report that explains your analysis about the visualization that you have done.**

The compressed file must include:

- Chart.py
- Date.py
- FileIO.py
- main.py
- TemperatureData.py
- WeatherAnalyzer.py
- **report.pdf**

Report Structure:

The report should follow the following sections:

1- Name of Group and members (In the case you are working as a group). Fig 5 shows a sample of how the first page of your report should look like.

**ENGG233 Final Project
Calgary Weather Analyzer
Milestone 1**

Group Name: Ucal

Members:

1. Erfan Aghaeekiasaraee (email: erfan.aghaeeaghae@ucalgary.ca)
2. Upma Gandhi (email: upma.gandhi@ucalgary.ca)

Lecture: Lecture#1

TA Group: Group Blue

Fig. 5

2- In the body of your report you must have 10 sections. In each section that is titled by a function you should write at least one paragraph that shows you understood what the result is showing.

Sections that your report must include:

1- Get Minimum Temperature of 1990-2019

Put the picture of your result and explain about that.

Example: According to provided Calgary weather data the minimum temperature between years 1990 and 2019 was xx temperature.

2- Get Maximum Temperature of 1990-2019

Put the picture of your result and explain about that.

3- Get Minimum Temperature of 1990-2019 Annually

Put the picture of your result and explain about that.

4- Get Maximum Temperature of 1990-2019 Annually

Put the picture of your result and explain about that.

5- Get Average Snowfall between 1990-2019 Annually

Put the picture of your result and explain about that.

6- Get Average Temperature between 1990-2019 Annually

Put the picture of your result and explain about that.

Example: According to the analysis that we have done on Calgary Weather data between 1990 and 2019. The average temperature shows a slight increase in 30 years and the maximum pick was in year xx.

7- LineChart Minimum Temperature of 1990-2019 Annually

Put the picture of your result and explain about that.

Example: According to the visualization that we have done on Calgary Weather data between 1990 and 2019. Always the temperature is lower than xx degrees. Calgary has experienced the lowest temperature in xx year which was xx degree.

8- LineChart Maximum Temperature of 1990-2019 Annually

Put the picture of your result and explain about that.

9- BarChart Average Snowfall between 1990-2019 Annually

Put the picture of your result and explain about that.

10- BarChart Average Temperature between 1990-2019 Annually

Put the picture of your result and explain about that.