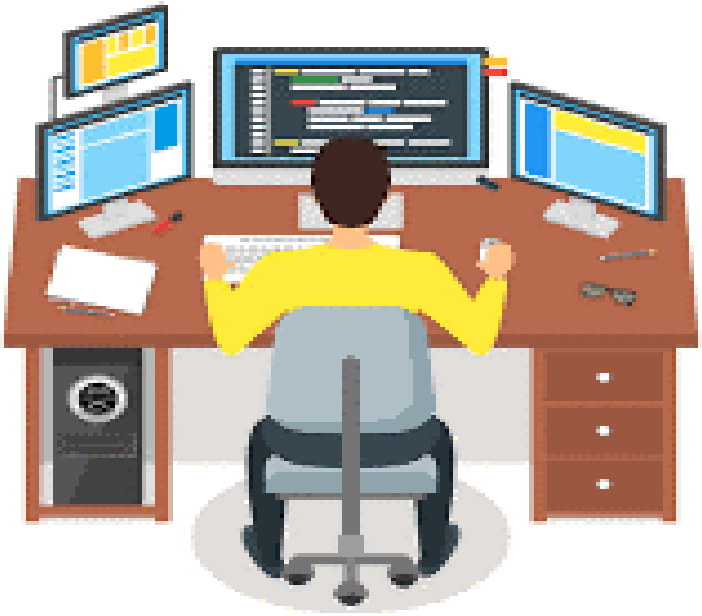
Midterm Sprint Week

June 16 – 28, 2023



- **Each project is to be completed in your groups.**

- **All projects are due by Wednesday June 28 by 11:59 pm. Each project is to be submitted through the assignment portal. Submissions are final – once you submit there is no resubmissions allowed. One member of the group should submit all projects.**

- **There are no lectures during Sprint Week. I will be on-line through Teams at 1:00 pm each day for questions or issues.**

- **I expect that Groups will have a Planning Meeting to set up the Trello Board, morning Stand-Up Meetings to go over the plan for the day and a Retrospective Meeting at the end to review the process, what did you do well, what could be improved.**

Project 1 – Python Program

This Python program will be based on the claims processing system for employees who take work trips for the NL Chocolate Company based on the following guidelines.

1. Start by presenting the following menu to the user. Validate the input to allow only 1 – 5.

NL Chocolate Company
Travel Claims Processing System

1. Enter an Employee Travel Claim.
2. Fun Interview Question.
3. Cool Stuff with Strings and Dates.
4. Something Old, Something New
5. Quit Program.

Enter choice (1-5):

In addition to the functions for the menu options, include at least **2 other functions** in the program that accept values as parameters and return a value(s) to the program – one for a calculation and one for another purpose – I will accept the FormatValues.py module as long as you add something to it. Do not add a new option to the menu. Add comment to all functions including a description, list of parameters, and a list of return values.

2. Option 1 – Enter an Employee Travel Claim.

The **NL Chocolate Company** needs a program to process salesperson travel claims when they return from a business trip. As employees return from business trips, they record all required information on a Travel Claim Form, and return the form, with all invoices, to the main office.

Allow the user to enter information from the Travel Claim Form including the employee number (must be entered and be 5 characters), employee first name (must be entered – adjust to title case), last name (must be entered – adjust to title case), location of the trip, start date (Must be entered and valid), end date (Must be entered and after the start date by no more than 7 days), the number of days (Must be entered between 1 and 7) (**BONUS: rather than input the number of days use the start and end dates to calculate this value)**, a value to indicate if they used their own car, or if a car was rented (Must be entered and must be the letters O or R only). Adjust to upper case), the total kilometers traveled (Must be entered and cannot exceed 2000). Only enter the kilometers if the employee used their own car, and the claim type as standard or executive (S/E) (Must be S or E – adjust to uppercase). Only **two** validations are required.

Calculate the per diem amount by multiplying the total days by a daily rate of 85.00 per day. The mileage amount is calculated using a rate of .17 per kilometer if the salesperson used their car, or a rate of $65.00 per day is the salesperson rented a car. The bonus is calculated based on the following:

If the number of days is more than 3 add $100.00 to the bonus. If the kilometers are over 1000 and the salesperson used their own car, add an extra 4 cents per kilometer to the bonus. If the claim type is Executive add an extra $45.00 per day to the bonus. Finally, if the start date is between Dec 15 and Dec 22 add an extra $50.00 per day to the bonus.

The Claim Amount is calculated as the per diem amount, the mileage amount, and the bonus. The HST is calculated on the Claim Amount using a rate of 15%. The Claim Total is the Claim Amount plus the HST.

The program will display all input and calculated values to the screen as results. Only display the mileage amount if it is calculated. Just do a basic printout with headings and formatted values.

Repeat the program until the user enters a termination value either on the first input, or a prompt at the end of the claim output.

3. **Option 2 – Fun Interview Question**

A common program used at interviews for programming position is the FizzBiss problem.

Create a loop to execute 100 times. For each value if the number is divisible by 5 display the word Fizz. If the value is divisible by 8 display the word Buzz. If the value is divisible by both 5 and 8 display the word FizzBizz. Otherwise display just the number. A sample of the output is shown. Display a message at the end of the option to "Press any key to continue …".

```
1
2
3
4
Fizz
6
7
Bizz
:
:
38
39
FizzBizz
41
:
```

4. **Option 3 – Cool Stuff with Strings and Dates**

Keeping with the theme of the program, create a program that does something cool with strings and dates based on the employee first name, last name, phone number, current date, the employee's start date, and the employee's birthdate. You can input the values or just assign values to variables. Display a message at the end of the option to "Press any key to continue …".

5. **Option 4 - Something Old, Something New**

Research something new in Python – could be a new library or process. Show how that topic is used by setting up an example. Include comments in the code to tell me what you are doing. Display a message at the end of the option to "Press any key to continue …".

Project 2 – Technical Writing

Prepare a user's manual for the program you wrote in Project 1. For the user, assume the program is installed on their system and there is an icon on the desktop. Include a cover page, a table of contents, and several of the items in the table of contents expanded in the instruction. You do not need to explain every option, but at least 3 should be expanded. Columns, pictures, and cool features are nice. You can save this as a pdf to keep your features in proper order.

Include your group # and team members on the cover page.

Project 3 – Bash – Prepare the following in a document.

Write the command(s) to complete each of the following in a document. You can test in Bash to check that everything is OK – but all I want is the document at the end.

- Create 3 files – one called **PySample.py**, one called **ExtraP.txt**, and a third called **Stuff.py**. Add code to each python program and add some text to txt file. Add the words **"RetailCost"** and **"getName"** in one or more of these files.

- What is the current directory? List the contents of the current directory. List the contents of the current directory with any hidden files. List the contents of the current directory with any hidden files. List the files with the permissions displayed.

- Display the full contents of each of the files created above. Display only the first 3 lines of **PySample.py** and the last 3 lines of **Stuff.py**. Display the last 3 lines of all files. If you have a file that is large, what options are available to have it appear one screen at a time?

- Create a directory called **PythonCurrent**, one called **PythonDone** and one called **TextFiles**. In the PythonDone directory create 2 other directories called **2022** and **2023**. In the **TextFiles** directory create 2 other directories called **Python** and **Other**.

- Change the current directory to **PythonDone – notice how the prompt changes to show the current directory**. Use pwd to confirm you are in the PythonDone directory. List the files.

- Change to the following directories: **PythonCurrent**, **2023**, and **Other**. Prove that you are in the proper directory. Go back to your working directory.

- Move the **Stuff.py** file to the **PythonCurrent** folder and give it the same name. Check and make sure the file has been moved. Copy the **ExtraP.txt** file to the **Python** folder in **TextFiles**.

- Change to the **Python** folder and display the first 6 lines of the file **ExtraP.txt**. Once complete move back to the main folder.

- Display the contents of the **ExtraP.txt** file from the Python directory in **PythonDone** directory from the current location – you should be in the working directory.

- From the working directory, find the file called **Stuff.py** searching all subdirectories.

- Change the permissions in the PySample.py so that the owner, group, and everyone else has only read and execute permission.

- Find the text **getName** in a file and indicate which file(s) it is located –search all subdirectories. Do the same with the word **RetailCost** – in this case allow the search by ignoring case. Direct the output for one of these to a file called **FindResults.txt**. How could you send both results to the same file without overwriting it?

- Perform one other command that you feel would be useful. Write it down with an explanation indicating what it is doing and why you feel it would be beneficial.

Project 4 – Presentation – to be presented on Friday Jun 23 @ 1 pm.

Each member of the group – not just one topic for the entire group - must find an article using an IT News site like https://www.pcmag.com/news, https://www.zdnet.com/, or https://www.computerworld.com/news/. Once you find your article, do a bit of research to find more information about the topic selected. I would strongly suggest that you find an IT news site you like and follow it regularly.

Create 2 - 3 slides based on your article and the research that you did. The first slide should summarize your article and the other 2 slides should be based on additional information that you researched.

Put all the slides from your group together with a consistent format. The first slide will be a title, the second slide will introduce the group members and their topics, and the remaining slides will be the ones you each prepared about your topic.

On Friday afternoon at 1 pm you will present your findings to the class. One member of the group will introduce the first two slides, then each member of the group should present their slides. The presentation should last approximately 5 minutes for each group.

Your presentation is to be included as part of your final submission.

Project 5 – Web Design

You are to design and create a web site on the topic of your choice. The site is required to contain at least **five (5) pages**. Your site will be graded for **originality** and **design consistency**, as well as efficient use of the following:

- **Page Layout –** make the layout easy to read and navigate. Keep the KISS rule in mind – Keep It Simple Stupid. Make proper use of semantic tags and style each appropriately.

- **CSS –** Create the page layout using CSS and the Box Model. Use of margins, borders and padding as required for attractive and effective design. Common styles and classes – especially text and links – are to be defined in CSS file and used for each page in your site.

- **Text formatting** – fonts, sizes, colors, bold, italic, underline, etc.

- **Use of colors** - for backgrounds and text.

- **Use of links** – for navigation on the top of each page, at least two (2) links to related external sites, and Top of Page links for pages that exceed the browser screen.

- **Use of images** – at least five (5) related images, not including those on the main page.

- **Use of lists** – at least two lists, ordered or unordered.

- **Use of tables** – one (1) table for tabular data.

You may create a banner for your site using any program you wish – but it must be saved as a .jpg file. The remainder of this site is to be completed using **HTML / CSS only**. Note that your site will be graded using Google Chrome.