**Reflection on Fundamentals of Python**

1. **In the past week(s) you have learned (i.e., researched, read, applied) new aspects of your  
   chosen programming language**

* **What did I learn about python**

Over the past few weeks, I have gained valuable insights into Python while working on my first project. Some key aspects I have learned include:

1. Object-Oriented Programming (OOP):

I explored how to structure code using classes and objects in python, which helped

in organizing data efficiently and improving code reusability.

1. File Handling:

I learned how to read and process data from external files, such as CSV files, to extract meaningful information and work with structured data.

1. Data Structures:

I gained experience in using built-in data structures to store and manage data effectively, for example list data structure can store different data types and I can used inbuild methods to allow easy access and manipulation of records.

1. Loops and Control Flow:

I implemented looping structures to iterate through data and perform operations efficiently while maintaining control over program execution.

1. Exception Handling:

I understood how to implement of handling unexpected errors in Python for a smooth program execution. Its simpler than other OOP languages.

1. Commenting and Documentation:

I learned two types of commenting in Python, writing clear and concise comments helped me in making the code more understandable for others and future reference.

* **What was interesting or fun about python learning**

I found Python’s simplicity and readability to be the most enjoyable aspects of learning. Unlike other programming languages I have experience with, such as C and Java, Python’s straightforward syntax made writing code faster and easier to understand. Additionally, working with error handling and file operations was particularly rewarding, as it allowed me to create a more reliable and user-friendly program.

1. **With regard to learning resources, that you used to learn (books, videos, web resources, other programmers etc.)**

To learn and apply Python concepts, I used various learning resources such as online documentation, YouTube tutorials, chatbots and forums.

* **Resources That Worked Best for Me:**

Python Documentation and W3Schools: These were the most helpful because they provided clear explanations and step-by-step examples, making it easy to understand and apply new concepts.

Chatbots and Forums (ChatGPT, Stack Overflow): They were useful for quick troubleshooting and clarifying specific doubts during coding.

* **Resources That Were Less Successful:**

Some YouTube Videos: Although useful, some were too lengthy and covered broad topics that were not relevant to my specific project needs.

Long Articles and Blogs: They contained too much information at once, making it challenging to quickly find what I needed.

* **Time-Consuming vs. Quick Resources:**

Most Time-Consuming: Python Documentation and YouTube tutorials with the examples

Least Time-Consuming: W3Schools on Python for example base learning and ChatGPT for interactive features so I can clear my doubts and comparing my previous knowledge of C and Jave and how its different implementing in Python

* **Effective Resources Based on My Experience:**

Python Documentation for in-depth learning.

W3Schools for easy-to-follow tutorials.

ChatGPT and Stack Overflow for fast problem-solving.

1. **In Research Assignment 1, I have created a Work Breakdown Structure and Gantt chart**

* **WBS accuracy**

I used my professor's WBS structure example as a guide and worked on to fit my project needs. It provided a clear outline and helped me stay on track. However, I realized that some tasks needed more detail, especially for testing and debugging. In the future, I will focus on breaking down tasks further to better manage my time and workload.

* **Lessons learned on WBS for future work**

In future projects, I will ensure that I include smaller, more detailed tasks and allocate time for potential challenges such as debugging and documentation.

* **Time estimation in Gantt chart**

I found that I underestimated the time required for testing and refining my code. Some tasks, such as learning file handling and exception management, took longer than expected.

* **Improvement on time estimation**

To improve, I will analyze my past projects and plan buffer time for potential roadblocks. I will also rely on my experience to make more realistic estimates in future assignments.