**Cover Page**

Full Name: Meet Maheta

Course Name: CST8333

Assignment Title: Programming Language Research Project - Reflective Report 1

Submission Date: June 21, 2024

**What was learned about my programming language**

**Programming Language: Python**

**Topics Learned:**

1. Basics of Python syntax and semantics
2. Data structures: lists, dictionaries, tuples, and sets
3. Control flow: loops and conditional statements
4. Functions and modules
5. File handling
6. Exception handling
7. Object-Oriented Programming (OOP)
8. Libraries and frameworks: NumPy, Pandas, Tkinter

**Strengths of Python:**

* **Ease of Learning and Use:** Python's syntax is straightforward and similar to English, making it easy to learn and use.
* **Extensive Libraries:** Python has a vast standard library and numerous third-party libraries, enabling quick and efficient development.
* **Versatility:** Python can be used for web development, data analysis, artificial intelligence, scientific computing, and more.
* **Community Support:** Python has a large and active community, providing extensive documentation, tutorials, and forums for support.

**Weaknesses of Python:**

* **Performance:** Python is slower compared to compiled languages like C or C++ due to its interpreted nature.
* **Mobile Development:** Python is not commonly used for mobile app development, with limited frameworks and support.
* **Memory Consumption:** Python can be memory-intensive, which might be a limitation for resource-constrained environments.

**The Best Resources for Me to Learn Are**

**Resource Table:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | ***Resource*** |  |  | | --- | |  | | |  | | --- | | ***Time Consumption*** |  |  | | --- | |  | | |  | | --- | | ***Usefulness*** |  |  | | --- | |  | | | ***Rank*** | | --- |  |  | | --- | |  | |
| Books | High | Medium | 3 |
| Websites | Medium | High | 2 |
| API Documentation | Low | High | 1 |
| Other Programmers | Medium | High | 2 |
| Tutorial Videos | Low | Medium | 3 |
| Online Training Courses | Medium | High | 2 |

**Most Effective Resources:**

I found API documentation to be the most effective due to its comprehensive and detailed nature, allowing me to understand the functions and methods in Python thoroughly.

Websites and online training courses are also highly effective because they provide a range of examples and interactive learning experiences. Books, while thorough, can be time-consuming to go through.

**WBS, Project Management Software, Reflection on Time Estimation**

**Reflection on Time Estimation:**

In the first half of the course, I found that my time estimation varied. For some tasks, I underestimated the time required, particularly for debugging and understanding complex concepts.

For others, I overestimated, especially for simpler coding tasks. Moving forward, I plan to break down tasks into smaller, more manageable components and allocate buffer time to account for unforeseen issues.

This approach should help improve my accuracy in time estimation.

**Discussion Board Post Archive**

**Discussion Forum 1 Post:**

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

**What did you learn about your programming language? (provide a list)**

I have learned the following about Python programming:

Python IDE - VScode: Figuring out how to navigate and use VScode to make coding easier.

Python Basic Syntax: Reinforcing the fundamentals of Python syntax and structure.

Reading CSV Files in Python: Learning how to read CSV files and store the data into containers like lists and pandas DataFrames.

Python Libraries: Exploring useful libraries like NumPy and Matplotlib for data analysis and visualization.

**What was interesting or fun about what you have learned, why? (briefly discuss)**

Python Syntax: Python's syntax is clean and readable, making it enjoyable to write and understand code quickly. Its simplicity allows for rapid development, which is very rewarding.

Using PyCharm: Exploring PyCharm was pretty cool. It has fantastic features like code completion and debugging that made my coding sessions smoother. Plus, the integration with version control systems made project management easier and more fun.

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

**Reflect on: What one(s) worked best for you, and which one(s) was (were) less successful for you and why?**

Best Resources

Web Resources: Sites like Real Python and the official Python documentation were friendly to beginners and I could quickly find the info I needed.

Books: "Automate the Boring Stuff with Python" by Al Sweigart was excellent for practical applications and learning best practices.

Less successful Resources

YouTube Tutorials: Some tutorials were great, but others weren't as detailed or clear, and they took a lot of time to watch.

Forums: Sometimes the answers were fragmented or too complex, which made it harder to grasp the full picture.

Reflect on: What one(s) was (were) the most time consuming, and what one(s) was (were) the least time consuming and why?

Most Time Consuming: YouTube tutorials and books took the most time because of the need to watch long videos or read through chapters.

Least Time Consuming: Web resources were quick and direct, saving me a lot of time.

**List your resources, and indicate what ones were the most effective for you (based on personal preference and time consumption to use).**

Web Resources:

"Python Documentation." Python Software Foundation. Available: https://docs.python.org/3/. (accessed May 14, 2024).

"Real Python." Real Python. Available: https://realpython.com/. (accessed May 14, 2024).

"PyCharm Documentation." JetBrains. Available: https://www.jetbrains.com/pycharm/documentation/. (accessed May 14, 2024).

Books:

A. Sweigart, Automate the Boring Stuff with Python: Practical Programming for Total Beginners. San Francisco, CA, USA: No Starch Press, 2015.

YouTube Tutorials:

Corey Schafer, USA. Python. [Online Video]. Available: https://www.youtube.com/user/schafer5. (accessed May 14, 2024).

Most Effective for Me: Web resources, especially the official Python documentation and Real Python, because they gave clear and detailed explanations quickly.

**3) In Research Assignment 1, you created a Work Breakdown Structure and Gantt chart**

**Reflect on: Was your WBS accurate, i.e., was it too high level, or too detailed?**

My WBS is accurate and served its purpose for the project phase 1.

**Reflect on: How will you improve your process when creating a new WBS in future?**

In the future, I plan to improve my WBS process by comparing the breakdown of tasks in WBS with my real work procedures for phase 1, and create a more refined and effective WBS for the future works.

**Reflect on: How was your time estimation on your Gantt chart? I.e. over-estimated, underestimated.**

My Gantt chart was generally effective, but there were instances where my time estimation was off. Some tasks took longer than expected, while others were completed more quickly. Overall, it provided a good visual representation of the project timeline, but there's room for improvement in estimating task durations more accurately.

**Reflect on: How will you improve your time estimation in future?**

I plan to keep a journal of task durations, comparing estimated times with actual times to refine my estimation skills. Additionally, discussing timelines with classmates and seeking advice from experienced individuals will help create more accurate future estimates.

**Discussion Forum 2 Post:**

**What I Learned About My Programming Language:**

In the past few weeks, I have delved deeper into various aspects of Python, enhancing my understanding and application of the language. Here's a summary of my learnings:

MVC Pattern: Previously, I had learned about the MVC (Model-View-Controller) pattern in Java. Recently, I successfully implemented this pattern in Python for my Practical Project 2.

Program Architecture: Creating UML diagrams and other architectural diagrams has been invaluable. These visual tools allow me to see the program's structure at a glance, making it easier to understand and organize the code.

**Interesting and Fun Aspects of My Learning:**

The most interesting and enjoyable part of my recent learning journey has been adding new functionalities to my projects, such as creating, selecting, and editing records. Implementing these features, especially integrating new functionalities into dropdown menus and developing new methods, was both challenging and fun.

**Learning Resources:**

Reflecting on the learning resources I've used, here are my insights:

**Most Effective Resources:**

YouTube Video Tutorials: These have been the most effective for me. They offer a great balance between depth and time investment. For example:

Learn Python - Full Course for Beginners: YouTube Link

Python for Beginners - Learn Python in 1 Hour: YouTube Link

Online Resources: The Python official site (python.org) has been a reliable resource for comprehensive and up-to-date information.

**Less Successful Resources:**

Books: While books like "Python Crash Course" by Eric Matthes provide thorough explanations, they are time-consuming and less engaging for me compared to video tutorials.

**Time Consumption of Resources:**

Most Time-Consuming: Reading books. This method requires extensive reading and note-taking, which slows down the learning process.

Least Time-Consuming: Watching video tutorials on YouTube. They offer condensed information that speeds up my understanding and retention.

**Work Breakdown Structure (WBS) and Gantt Chart Reflections:**

Accuracy of WBS: Initially, my WBS plan was not followed strictly, but I adapted and found that my WBS for Practical Project Part 2 was quite accurate.

Improving Future WBS: I aim to make future WBS plans more accessible and precise. While perfection isn't guaranteed, continuous improvement is the goal.

Gantt Chart Time Estimation: Initially, using Microsoft Project to create Gantt charts was challenging due to unfamiliarity. However, with practice, it has become much easier.

**Future Improvements:**

WBS: I will strive to make WBS plans more detailed and accurate, ensuring they are practical and achievable.

Time Estimation: To improve time estimation, I plan to break down larger tasks into smaller, manageable chunks, allowing for more precise and efficient time management.

# **References:**

|  |  |
| --- | --- |
| [1] | G. v. Rossum, “Python Programming Language,” 1991. [Online]. Available: https://www.python.org/. |
| [2] | M. Lutz, “Learning Python, O'Reilly Media, Inc.,” 2013. [Online]. |
| [3] | T. D. Science, “N-Layer Architecture in Python,” Towards Data Science, 2021. [Online]. Available: https://towardsdatascience.com/n-layer-architecture-in-python-4eccfa3b1fae. [Accessed 30 May 2024]. |
| [4] | “Real Python,” Real Python, [Online]. Available: https://realpython.com/understanding-model-view-controller/. [Accessed 30 May 2024]. |
| [5] | D. S. Foundation, “Django Documentation,” [Online]. Available: https://docs.djangoproject.com/en/stable/. [Accessed 29 May 2024]. |
| [6] | P. Projects, “Flask Documentation,” [Online]. Available: https://flask.palletsprojects.com/en/latest/. [Accessed 29 May 2024]. |