**Cover Page**

Full Name: Meet Maheta

Course Name: CST8333

Assignment Title: Programming Language Research Project - Reflective Report 2

Submission Date: August 9, 2024

**What was learned about my programming language**

**Programming Language:** **Python**

### **Topics Learned:**

1. **Advanced Python Syntax and Semantics**
2. **Data Structures: Advanced usage of lists, dictionaries, tuples, and sets**
3. **Control Flow: Mastery of loops and conditional statements**
4. **Functions, Modules, and Packages**
5. **File Handling: Reading from and writing to various file formats (CSV, JSON, XML)**
6. **Exception Handling: Advanced techniques and custom exception classes**
7. **Object-Oriented Programming (OOP): Inheritance, polymorphism, and encapsulation**
8. **Libraries and Frameworks: Advanced usage of NumPy, Pandas, Tkinter, Flask, Django**
9. **Web Development: Building and deploying web applications using Flask and Django**
10. **Data Analysis and Visualization: Using Matplotlib, Seaborn, and Plotly**
11. **Testing: Writing unit tests with PyTest and integrating Continuous Integration (CI) tools**

### **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 3 Post:**

### Learning Aspects of Chosen Programming Language

In the past week(s), I have researched, read, and applied new aspects of Python. Here is what I have learned:

1. Learned how to implement polymorphism in Python.
2. Explored how polymorphism enhances code flexibility and reuse.
3. Learned about database connectivity using MySQL in Python.
4. Implemented CRUD operations using MySQL within a Python application.
5. Learned how to write and run unit tests for database interactions in Python.

### Interesting and Fun Aspects

The most interesting topic was learning about polymorphism. It was fascinating to see how it makes code more modular and adaptable, allowing objects to be treated as instances of their parent class rather than their actual class. Additionally, implementing database connectivity with MySQL was rewarding as it provided hands-on experience with managing data persistence in applications.

### Learning Resources

**Effective Resources:**

* **YouTube Videos:** I found video resources to be the most useful. YouTube provided both visual and auditory learning inputs, which helped reinforce concepts.
  + [How Polymorphism Works in Python](https://www.youtube.com/watch?v=cyP4Uw2b2XM&t=2s) (Code Stack)
  + [Learn MySQL in Python](https://www.youtube.com/watch?v=nGIg40xs9e4&t=103s) (Tech With Tim)
  + [Using MySQL with Python](https://www.youtube.com/watch?v=vs6dXL9Wp7s&t=6s) (Very Academy)
* **Official Documentation:** The MySQL command line interface documentation was particularly helpful for understanding specific commands and functionalities.
  + [Python MySQL Documentation](https://dev.mysql.com/doc/connector-python/en/)

**Less Successful Resources:**

* General documentation was less helpful compared to specific CLI documentation. It was more time-consuming to sift through extensive documentation to find relevant information.

### Time Consumption of Learning Resources

* **Most Time Consuming:** Video resources were the most time-consuming because I tend to rewatch content to ensure a full understanding of the material.
* **Least Time Consuming:** Documentation was less time-consuming overall, though it can be time-consuming if relied upon extensively.

### List of Resources by Effectiveness and Personal Preference

1. [How Polymorphism Works in Python](https://www.youtube.com/watch?v=cyP4Uw2b2XM&t=2s) (Code Stack)
2. [Learn MySQL in Python](https://www.youtube.com/watch?v=nGIg40xs9e4&t=103s) (Tech With Tim)
3. [Using MySQL with Python](https://www.youtube.com/watch?v=vs6dXL9Wp7s&t=6s) (Very Academy)
4. [Python MySQL Documentation](https://dev.mysql.com/doc/connector-python/en/)

# 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 29 June 2024]. |
| [4] | R. Python, “Real Python,” [Online]. Available: https://realpython.com/understanding-model-view-controller/. [Accessed 29 June 2024]. |
| [5] | D. S. Foundation, “Django Documentation,” [Online]. Available: https://docs.djangoproject.com/en/stable/. [Accessed 30 June 2024]. |
| [6] | P. Projects, “Flask Documentation,” [Online]. Available: https://flask.palletsprojects.com/en/latest/. [Accessed 30 June 2024]. |