|  |  |
| --- | --- |
| Algonquin College – Co-op – Research Assignment 1 | |
| Programming Language Research and Tools | |
| 25W\_CST8002\_040 Programming Language Research Projects |

|  |
| --- |
| Student: Vaishali Jaiswal  Due: 1-19-2025 |

Table of Contents

[**1. Language Research** 2](#_Toc188215954)

[**2. Language Selection** 5](#_Toc188215955)

[**3. Unit Testing Research** 6](#_Toc188215956)

[**4. Development Tools: Platform, Tool Availability, Licensing, Version Control** 7](#_Toc188215957)

[**5. WBS and Gantt Chart for Practical Project Part 1** 8](#_Toc188215958)

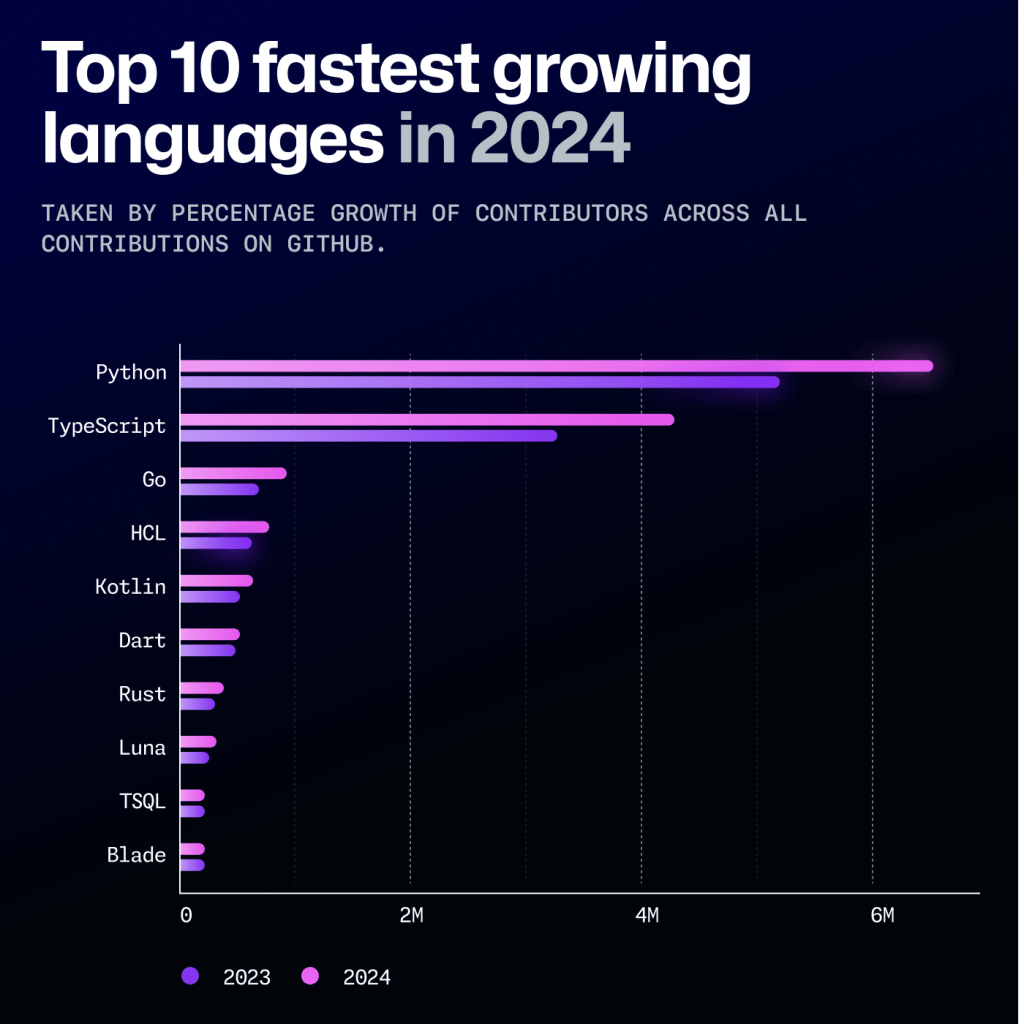
[**6. Confirmation of Data Set for Use** 9](#_Toc188215959)

[**7. References** 10](#_Toc188215960)

## **1. Language Research**

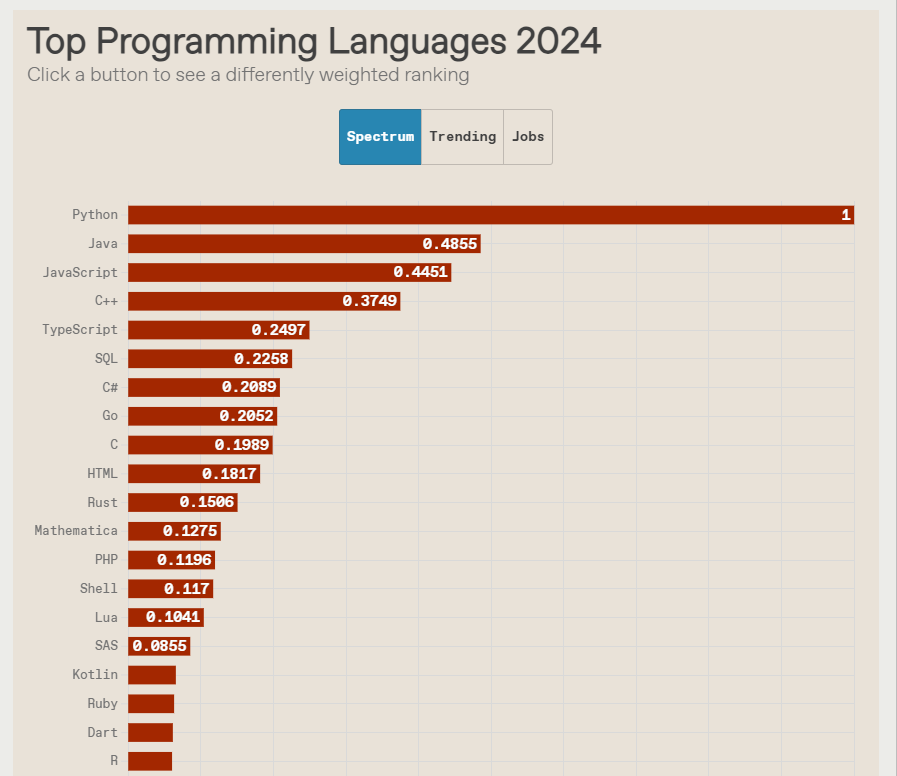
#### **Websites Listing and Ranking Popular Programming Languages:**

1. **GitHub Octoverse -** <https://octoverse.github.com>



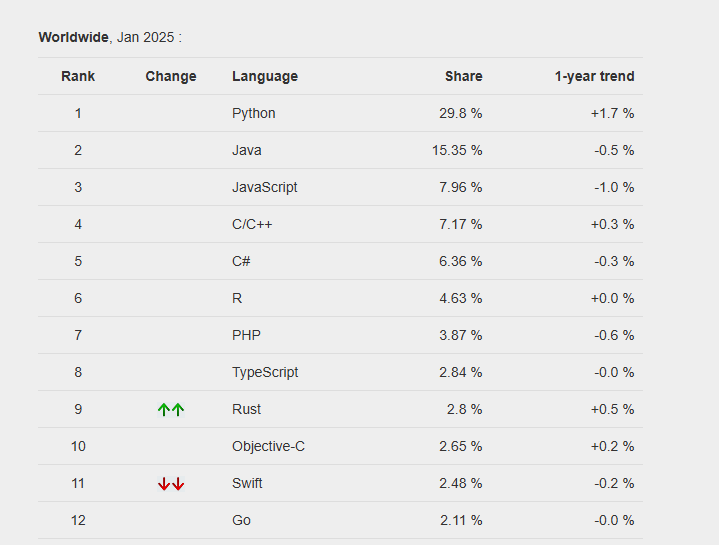
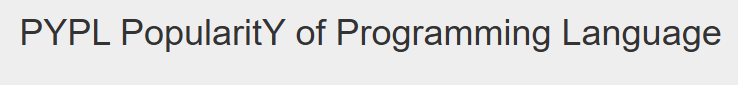
* Provides insights into the most popular programming languages based on repository activity and developer contributions.
* Reference: GitHub, "The State of the Octoverse," 2024. [Online]. Available: https://octoverse.github.com/ [Accessed: 5 Jan. 2025]
* Reputable as it collects data from millions of developers worldwide.

1. **IEEE Spectrum Ranking -** <https://spectrum.ieee.org/top-programming-languages>



* Ranks programming languages based on multiple criteria including search trends, job postings, and online discussions.
* Reference: IEEE Spectrum, "Top Programming Languages 2024," 2024. [Online]. Available: https://spectrum.ieee.org/top-programming-languages [Accessed: 5 Jan. 2025]
* Reputable due to its data-driven approach and technical analysis.

1. **PYPL Index -** <https://pypl.github.io/PYPL.html>



* Measures the popularity of programming languages based on Google search frequency for tutorials.
* Reference: PYPL, "PopularitY of Programming Language index," 2024. [Online]. Available: <https://pypl.github.io/PYPL.html> [Accessed: 5 Jan. 2025]
* Reputable as it reflects real-time learning interest trends.

## **2. Language Selection**

**Selected Language:**

* Python

**Reasons for Selection:**

* Widely used in multiple fields such as data analysis, web development, and automation.
* Large community support and availability of resources for learning and development.

**Alignment with Career Goals:**

* Useful for building accessible applications and improving software development skills.

## **3. Unit Testing Research**

#### **Websites Detailing Framework-Based Unit Testing in Python:**

1. **PyTest Documentation**

* Provides an official guide and examples for writing unit tests in Python using the pytest framework.
* Reference: PyTest, "pytest documentation," 2024. [Online]. Available: <https://docs.pytest.org/en/stable>/ [Accessed: 5 Jan. 2025]

1. **GeeksforGeeks Python Testing Guide**

* Offers comprehensive tutorials on unit testing in Python with unit test and pytest frameworks.
* Reference: GeeksforGeeks, "Python Unit Testing Tutorial," 2024. [Online]. Available: <https://www.geeksforgeeks.org/unit-testing-python-unittest>/ [Accessed: 5 Jan. 2025]

**Selected Unit Test Framework:** pytest

* Chosen for its simplicity, detailed assertions, and wide adoption in the Python community.

## **4. Development Tools: Platform, Tool Availability, Licensing, Version Control**

**Development Environment:**

1. **Operating System:** Windows 11 Pro Version 10.0.22631, Build 22631
2. **Programming Language:** Python 3.12 - <https://www.python.org/downloads/release/python-3120/>
3. **Integrated Development Environment (IDE):** PyCharm 2024.3 - [2024.3.1.1 - Windows (exe)](https://download.jetbrains.com/python/pycharm-community-2024.3.1.1.exe)
4. **Version Control:** Git version 2.48.1.windows.1 - <https://git-scm.com/downloads/win>

**Licensing:**

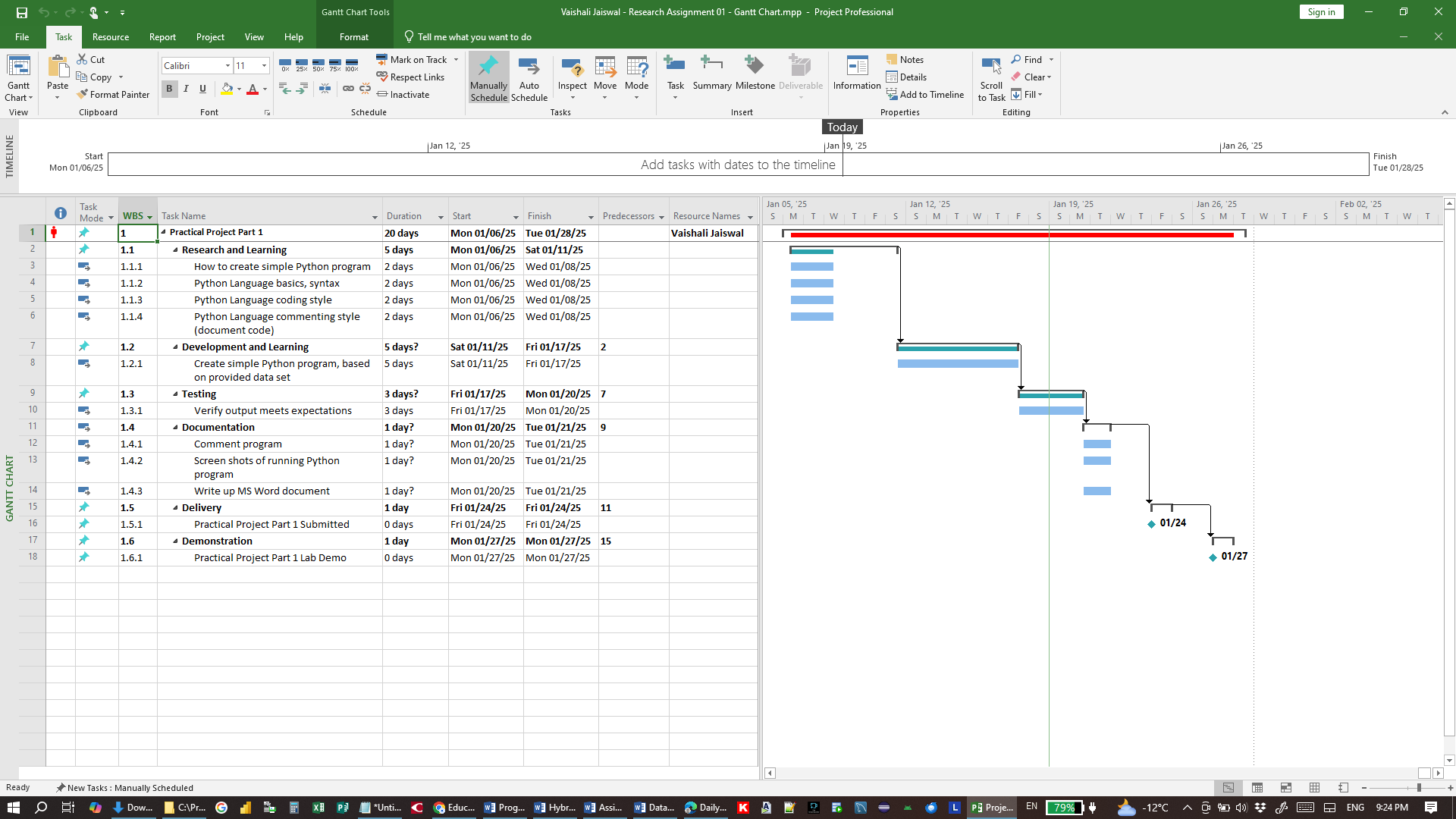
* All tools selected are free for personal use and available under open-source licenses.

## **5. WBS and Gantt Chart for Practical Project Part 1**

#### Work Breakdown Structure (WBS):

1. Practical Project Part 1
   1. Research and Learning
      1. How to create simple Python program
      2. Python Language basics, syntax
      3. Python Language coding style
      4. Python Language commenting style (document code)
   2. Development and Learning
      1. Create simple Python program, based on provided data set
   3. Testing
      1. Verify output meets expectations
   4. Documentation
      1. Comment program
      2. Screen shots of running Python program
      3. Write up MS Word document
   5. Delivery
      1. Practical Project Part 1 Submitted
   6. Demonstration
      1. Practical Project Part 1 Lab Demo

**Gantt Chart:** (Screenshot of Gantt chart)



## **6. Confirmation of Data Set for Use**

* **Data Set File Name:** dailyvehiclesdownload.csv
* **Data Set Title:** Daily vehicles per KM by municipality - Government of Alberta

## **7. References**

[1] GitHub, "The State of the Octoverse," 2024. [Online]. Available: https://octoverse.github.com/ [Accessed: 5 Jan. 2025]

[2] IEEE Spectrum, "Top Programming Languages 2024," 2024. [Online]. Available: https://spectrum.ieee.org/top-programming-languages [Accessed: 5 Jan. 2025]

[3] PYPL, "PopularitY of Programming Language index," 2024. [Online]. Available: https://pypl.github.io/PYPL.html [Accessed: 5 Jan. 2025]

[4] PyTest, "pytest documentation," 2024. [Online]. Available: https://docs.pytest.org/en/stable/ [Accessed: 5 Jan. 2025]

[5] GeeksforGeeks, "Python Unit Testing Tutorial," 2024. [Online]. Available: https://www.geeksforgeeks.org/unit-testing-python-unittest/ [Accessed: 5 Jan. 2025]