**Introduction-**

**In this project, I will compare four programming languages based on their execution time, compilation time, memory usage, and ease of writing code. For this comparison, we will implement three programs in all four languages, and based on the results, we will interpret the data and draw conclusions.**

**Language chosen-**

Python-

Python is high level, general purpose language. It is generally considered an interpreted language, that is, code in python is executed line by line by an interpreter at runtime, rather than being fully compiled into machine code before execution.

Java-

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers "write once, run anywhere" (WORA). This means that compiled Java code can run on all platforms that support Java without the need for recompilation, making it highly versatile.

C-

C is a powerful and versatile general-purpose programming language known for its efficiency and ability to handle low-level memory manipulation. C offers a rich set of features, including various data types, operators, control structures, and built-in functions. It also provides low-level access to memory through pointers.

Golang-

Go, often referred to as Golang, is a statically typed, compiled programming language designed by Google engineers Robert Griesemer, Rob Pike, and Ken Thompson. Go (Golang) is a compiled language. This means that Go source code is translated directly into machine code by a compiler before execution, resulting in an executable binary file.

**Task Description-**

1st Program (Binomial Series)-

The Binomial Expansion code is used to calculate and display the terms of the expansion of an expression of the form

(a+b)^n

Useful in probability and statistics for calculating probabilities in binomial distributions.

2nd Program (Armstrong Number)-

An **Armstrong number** code is used to check whether a given number is equal to the sum of its own digits each raised to the power of the number of digits. It helps you understand the concepts of digit extraction, exponentiation, and summation — all fundamental programming skills. The logic can be extended to:  
• Finding all Armstrong numbers within a given range  
• Validating special numeric patterns in cryptography  
• Building mathematical puzzles and number-based games

3rd Program (Leap Year)-

A **Leap Year** program is used to determine whether a given year has 366 days instead of the usual 365, which happens when February has 29 days. It helps you learn conditional logic, modular arithmetic, and how to handle special calendar rules — important for working with dates in programming. The logic can be extended to:  
• Validating and correcting date inputs in applications  
• Building scheduling and calendar tools  
• Accurately calculating time spans across years in software

**Metrics measured-**

Execution Time-

Execution time is the amount of time it takes for a program (or a specific part of it) to run from start to finish after it starts executing.

Compilation time-

Compilation time is the amount of time a computer takes to translate source code (human-readable program) into machine code that the processor can run.

Memory Usage-

Memory usage while running the program means the amount of RAM your program is actively using at that exact moment during execution.

Ease of writing a program-

The ease of writing a program is usually referred to as a language’s programming simplicity or ease of use.

**Results-**

We will try to compare Execution time, compilation time, memory usage and ease of writing for all the 3 programs

**Binomial series**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Python** | **Java** | **Golang** | **C** |
| **Execution time** | 0.000121 secs | 0.024626 secs | 0.000061 secs | 0.00000021 secs |
| **Compilation time** | 0.002272 secs | 0.000000 secs | 0.00032 secs | 0.000061 secs |
| **Memory Usage** | 11.77 KB | 49.125 KB | 0.539 KB | 3612 KB |

If we compare the ease of writing, the order from the easiest to most complicate will be as follows-

Python is very easy to write.

Golang is more stricter than Python.

Java is more verbose than Python.

C is the most complicated when it comes to ease of writing. It’s very strict about syntax.

**Armstrong numbers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Python** | **Java** | **Golang** | **C** |
| **Execution time** | 0.001199 secs | 4.70 secs | 0.000031101 secs | 0.000000021secs |
| **Compilation time** | 0.000831 secs | 1.19 secs | 0.000324187 secs | 0.000072 secs |
| **Memory Usage** | 11.69 KB | 1966 KB | 0.507 KB | 3612 KB |

If we compare the ease of writing, the order from the easiest to most complicate will be as follows-

Python is very easy to write.

Golang is more stricter than Python.

Java is more verbose than Python.

C is the most complicated when it comes to ease of writing. It’s very strict about syntax.

**Leap Year**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Python** | **Java** | **Golang** | **C** |
| **Execution time** | 0.000076 secs | 3.1222 secs | 0.000320227 secs | 0.000032 secs |
| **Compilation time** | 0.000735 secs | 0.003779 secs | 0.00002525 secs | 0.000078 secs |
| **Memory Usage** | 11.84KB | 1477.25 KB | 0.5 KB | 2680 KB |

If we compare the ease of writing, the order from the easiest to most complicate will be as follows-

Python is very easy to write.

Golang is more stricter than Python.

Java is more verbose than Python.

C is the most complicated when it comes to ease of writing. It’s very strict about syntax.

**Conclusion-**

**Execution Time (Fastest to Slowest)**

* C → Go → Java → Python
* C is fastest because it is compiled to machine code and runs close to hardware.
* Python is slowest due to being an interpreted, dynamically typed language.

**Compilation Time (Fastest to Slowest)**

* Python → Go → C → Java
* Python has no compilation (interpreted).
* Go compiles very fast due to its lightweight compiler.
* Java and C require more compilation time, with Java typically being slower due to bytecode generation and verification.

**Memory Usage (Lowest to Highest)**

* C → Go → Java → Python
* C has the lowest memory usage as it allows manual memory control.
* Python consumes the most memory due to its interpreter overhead and dynamic features.

**Ease of Writing (Easiest to Hardest)**

* Python → Go → Java → C
* Python has simple, concise syntax.
* Go is straightforward but more verbose than Python.
* Java is more verbose with strict OOP structure.
* C is lowest-level and requires manual memory management.

**Final Conclusion**

* If **performance** is your top priority → **C** is the best.
* If you want a **balance between speed and ease of writing** → **Go** is ideal.
* If you need **enterprise-level OOP and portability** → **Java** is a good choice.
* If you want **quick development and simplicity** → **Python** is the winner.