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 (Maclaurian series)-

This code is used to **approximate** sin(x)\sin(x)sin(x) using its **Maclaurin series** expansion instead of the built-in math.sin() function. It shows how sin(x)\sin(x)sin(x) can be expressed as an infinite series. This is a key concept in calculus and numerical analysis. It is useful in **Big Data Analytics** or **Scientific Computing** for understanding how approximations work when direct computation isn't possible.

2nd Program (Palindrome)-

This palindrome number code is used to **check whether a given number reads the same forwards and backwards**. It teaches you how to reverse data and compare values — a basic but important programming skill. The logic can be extended to:

- Palindrome words and sentences
- Palindrome dates (like 02/02/2020)
- Palindrome checks without converting to strings (pure math approach)

3rd Program (Deep Recursion)-

The **deep recursion** code's main use is to demonstrate what happens when a function calls itself many times, going far deeper than typical recursion examples. This code shows how to increase the limit and what happens when you go deeper. It helps to understand memory usage and performance issues in recursion-heavy problems.

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

Maclaurian series

	Python	Java	Golang	C
Execution time	0.000108 secs	0.000178 secs	0.0000001 secs	0.00000025 secs
Compilation time	0.001570 secs	1.101679 secs	0.000325 secs	0.000083 secs
Memory Usage	11.77 KB	983.65 KB	0.0 KB	128 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.

Palindrome numbers

	Python	Java	Golang	C
Execution time	0.000629 secs	0.000010 sec	0.000026 secs	0.000000023 secs
Compilation time	0.000583 secs	0.882858 sec	0.000317 secs	0.000075 secs
Memory Usage	11.67 KB	0.0 KB	0.512 KB	0.0KB

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.

Deep recursion

	Python	Java	Golang	C
Execution time	0.606381 secs	0.794779 sec	0.000082 secs	0.000056 secs
Compilation time	0.000688 secs	0.007766 sec	0.000321 secs	0.000069 secs
Memory Usage	11.70KB	4.43 KB	0.0 KB	128 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.