Lab 1: Exploring Compilation, Interpretation, and Language Behavior *2 Members/Group* - Due: Feb 28, 2025 (EoD)

Objective:

- Understand how different programming languages handle compilation and execution.
- Analyze the impact of static vs dynamic typing and error handling.
- Compare performance and code structure across multiple languages.
- Develop structured reporting skills.

Preferred Environment: Linux

Tasks:

Task 1: Compilation & Execution

- Implement a 'Hello, Compiler World!' program in three different languages (C++, Java, Python).
- Compile and execute the programs.
- Identify differences in compilation commands, execution processes, and dependencies.

Task 2: Compiler vs Interpreter – Performance Test

- Implement a simple arithmetic calculator (+, -, *, \) in Python, C++, and Java.
- Measure & compare execution time by running it 100,000 times in a loop (random value and operator per iteration).

Task 3: Static vs Dynamic Typing – Detecting Errors

- Write a function to sort an array of numbers in Java and Python.
- Introduce errors by inserting string values and observe behavior.

Task 4: Error Handling & Debugging

- Write a program that divides a number by another in three languages.
- Introduce division by zero, non-numeric input, and empty input errors.

Task 5: Code Readability & Efficiency

- Implement matrix multiplication in **C** (not C++) and Python.
- Compare lines of code and execution time for different matrix sizes (see report for size).

Submission Instructions

All submissions must be in a **single zip file** (zip only) named strictly as follows: ID1_ID2_lab1.zip

Inside the zip file, the folder structure must be:

ID1_ID2_lab1

code (Contains source code files: ‘Program.cpp’, ‘Program.py’, ‘Program.java’, ‘Program.c’)

report.pdf (Final report in structured format)

Any deviation from this structure may result in a penalty or rejection.

Lab 1 Report: Compilation, Interpretation, and Language Behavior

Group Members:

(Write names & IDs)

Task 1: Compilation & Execution

Language	Compilation Command	Execution Command	Extra Setup Needed?
C++	g++ Program.cpp -o cpp.out	./cpp.out	----------
Java	javac Program.java	java Program	----------
Python	python3 Program.py	python3 Program.py	----------

Task 2: Performance Test

Language	Execution Time (100,000 calculations)
C++	Execution Time: 0.144111 seconds
Java	Execution Time: 0.256226933 seconds
Python	Execution Time: 0.392931 seconds

Task 3: Static vs Dynamic Typing

Language	Behavior When Mixing Data Types	Solution to Fix
Java	Error because the function can’t take both array of string and int.	Make 2 functions
Python	Error when there is 2 different type in the list. No error if list is a single type	Be careful when writing the code.

Task 4: Error Handling & Debugging

Error Type	C Behavior	Python Behavior	Java Behavior
Division by 0	Output : c.out: Division by zero	Will throw an exception	Will throw an exception
Missing parameters	Can't compile the program	Throw an Error and program stop	Can't compile
Wrong type of argument	Work because a char is a number. Will not work with string because thay are array	Throw an error and program stop	Can't compile

Task 5: Code Readability & Efficiency

Language	Lines of Code	Execution Time (10x10)	Execution Time (100x100)
C	14	Time: 0.000021	Time: 0.014472
Python	6	Execution Time: 0.000407 seconds	Execution Time: 0.298866 seconds

Final Learning Reflection (2-3 Sentences)

- One thing we learned from this lab is __ how language can be faster but unsafe when handling error (memory issue with C/C++ but not with Python/Java)__.
- One challenge we faced was __handling the memory for the matrix multiplication in C__, and we solved it by __ looking in the web trying different algorithm to handle different size of matrix__.