**Execution times**

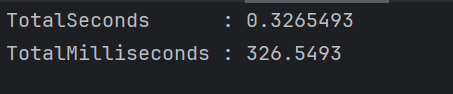
1

Execution time refers to the amount of time a program takes to complete. In the case of this code, it starts when the GUI has appeared on the screen and ends when the program is terminated. The execution time differs from programming language to programming language.

A screen shot of a computer

Description automatically generatedIn Python, the execution time is found below. It should be noted that to calculate the execution time for python , the program had to be manually terminated.

For Java, this was a different case, as there was no need for the program to be manually terminated.



From the results, it can be concluded that the Python’s execution time is vastly greater than that of Java (even with 2 seconds being deducted from pythons’ execution time to counter the error of human reaction time.).

**Differences due to programming language features**

Both java and python are popular programming languages which have their own unique features. These Differences involve dynamic vs static typing , GUI Implementations, and Object-Oriented Programming Principles.

* **Dynamic vs static Typing:** Python uses dynamic typing which allows the code to be more flexible as seen in “Main\_Updated.py” , where “game” is assigned an instance of “Game()”. Here, the variables can be assigned and re-assigned values of any type without any explicit alteration.

**A black background with blue and white text

Description automatically generated**

Java, on the other hand, requires strict type definitions (which enhances type security) but requires more boilerplate code, as seen in “Slider\_Game\_Model.java” which has explicit class and type declarations.

**A black screen with white text

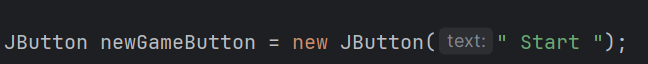
Description automatically generated**

* **GUI Implementation:** For Python’s GUI , libraries such as Pygame were used which allows for simple dynamic UI updates as can be observed in “Main\_Updated.py” .

**A black background with white text

Description automatically generated**

For Java , Java Swing is imported to “Slider\_GameGUI.java” , which is powerful for desktop applications but requires more verbose code to achieve similar UI functionalities to that of Python.



* **Object-Oriented Programming Principles:** Both languages support OOP Principles , but the way they are used is different. Python offers a less formal approach. Here the files demonstrate class definitions and instantiations and making use of encapsulation and inheritance, but, without the need for explicit interface annotations or even implementation.

**A computer code on a black background

Description automatically generated**

On the other hand, Java explicitly extends “container” in “ Slider\_GameGUI.java” and implements interfaces with “@override” annotations for method overriding, making use of inheritance and polymorphism.

****