```c
#include <stdio.h>
int main() {
   printf("Hello, World!");
   return 0;
}
```

Low-level v. High-level Programming Languages

```cpp
#include <iostream>

int main() {
    std::cout << "Hello World!";
    return 0;
}
```

BACKGROUND

- Programming languages are similar to human languages, except they use certain formats to make them computer readable. Each language is its own set of rules which give the computer instructions on what to-do.
- Higher level programming languages are languages that are easier to write and more human readable but, because of this, are harder to effectively & efficiently translate to a computer readable language
- Lower level programming languages are languages that are harder to write but less human readable, because of this, they allow fore more effective and efficient translation into a computer readable language
- Different languages are unique in the way they are written, just like any human language, they have different intricacies and instruct the computer in a different way to achieve the same results

PROCEDURE

1. *Initial setup:*
   a. *Ensure sure all languages & all needed Visual Studio Code Plugins are installed*
2. Record preliminary data including:
   - Size of runtime in megabytes and language runtime/compiler version number.
3. Run test measuring the time it takes of printing text to screen.
4. Record execution time using PowerShell measure command; Record to table.
5. Run test measuring time taken to calculate and print the result of various arithmetic:
   - Adding, Subtracting, Multiplying, and Dividing
6. Record execution time using PowerShell measure command; Record to table.
7. Run test measuring the time it takes to carry out conditionals:
   - if/else statement and switch statement
8. Record execution time using PowerShell measure command; Record to table.
9. Run test measuring the time it takes to carry out the execution of a loop statement, this being a for loop
10. Record execution time using PowerShell measure command; Record to table
11. Lastly run test and measure the time it takes to convert various datatypes:
    - Integer to string, and string to float.
12. Record execution time using PowerShell measure command; Record to table
13. Repeat steps 2 through 12 for every language being tested/compared
14. Cleanup:
    - Close Visual Studio Code, terminate PowerShell, and terminate any still open compiler, interpreters, and/or language VMs

**

Procedural Note(s):

For steps 7 & 9: Printing statements, which, print formatted result of multiplication arithmetic, will be executed within if/else statements, switch statements, and for loops.

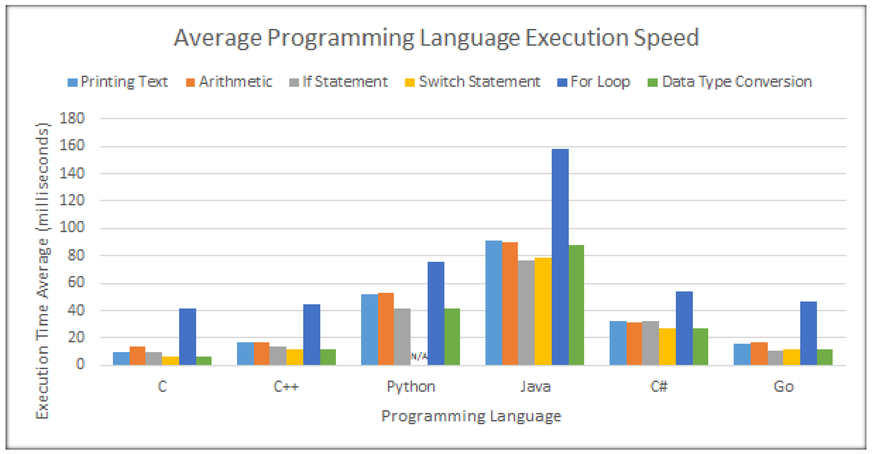
QUESTION

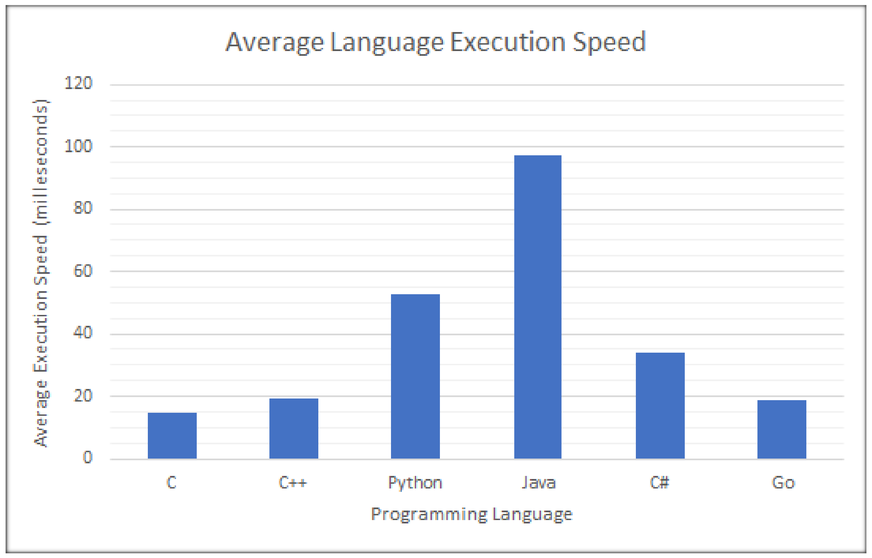
Do lower-level programming languages such as C, C++, and Go run/execute programs faster than higher-level languages such as Java, Python, and C#?

HYPOTHESIS

Lower level languages like C, C++, and Go will execute programs faster than higher level languages such as Java, Python, and C#.

RESULTS





ABSTRACT

The purpose of this study was to find whether higher-level or lower-level programming languages execute programs faster. The hypothesis of this study was that lower level languages like C, C++, and Go will execute programs faster than higher level languages such as Java, Python, and C#. The basic procedure of the study was to test speed of each programming language using a suite of tests ran against each language, including how fast they print text/variables to screen, how fast they do data-type conversion, how fast they run conditionals, et cetera. Each suite of tests was run 3 times and then averaged for each language. A major finding of this study was that the lowest-level language of C, was found to run about 6½ times faster than the highest-level language of Java. From the results of this study, its hypothesis was supported in finding that lower-level languages, overall, executed programs much faster than higher level languages. The results of this study can most likely be attributed to the fact that, by design, lower-level languages are closer to that spoken by a computer and therefore much easier to effectively and efficiently translate compared to the more human like higher-level languages.

CONCLUSIONS

- The tested lower-level languages of C, C++, and Go ran faster with average execution times of 14.5ms, 19.2ms, and 18.8ms respectively compared to the tested higher-level languages of Python, Java, and C# with average execution times of 52.9ms, 97.1ms, and 33.9ms respectively
- Results can most likely be attributed to the fact that, by design, lower-level languages are closer to that spoken by a computer and therefore much easier to effectively and efficiently translate and convert compared to the more human like higher-level languages
- Results of this experiment may have been affected by the way each program was written, meaning that, since programs written in each language are written syntactically differently, some programs may have been written more efficiently than others leading to skewed results

FUTURE DIRECTIONS

- In the future, I may expand this study to include more functions of each programming language and more complex tasks to improve coverage of all parts of the languages.
- In the future I may also expand this study to include more programming languages

Image Source: Unsplash; Link: https://unsplash.com/photos/rWKhlvk-bsM