




UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and
Information Technology

COS284

COMPUTER ORGANIZATION AND ARCHITECTURE

PRACTICAL 1

Name and Surname	Student Number	Signature
L.M. Burgess	18015001	

By submitting this assignment I confirm that I have read and am aware of the University of Pretoria's policy on academic dishonesty and plagiarism and I declare that the work submitted in this assignment is my own as delimited by the mentioned policies. I explicitly declare that no parts of this assignment have been copied from current or previous students' work or any other sources (including the internet), whether copyrighted or not. I understand that I will be subjected to disciplinary actions should it be found that the work I submit here does not comply with the said policies.

Contents

1	Introduction	2
2	Aim	2
3	Analysis Method	2
4	Results	2
5	Discussion of Results	3
6	Conclusion	3
A	System Specifications	4

List of Figures

1	Graph showing execution speed of different languages	3
2	System specifications	4
3	System specifications	4
4	System specifications	4

1 Introduction

The computer level hierarchy allows for different levels of instruction sets (software) to be implemented and run. Each level moving closer or further away from the hardware architecture comprising the computer system. As the designer moves away from the hardware level in the hierarchy, possible performance and optimisation to software written may be lost.

2 Aim

To investigate the performance and size difference of simple print statements in different programming languages. The programs being:

- Assembly
- C++
- Python
- Java

3 Analysis Method

Each statement has a basic print statement executed of:

The quick brown fox jumped over the lazy dog.

Where the execution time was measured across five-hundred runs and added together using a bash script, the UNIX command "time" and a python program for statistics computation. This was done a total of fifty times for each language to record and compute the average, maximum and minimum for each language. Specifications of the system used may be found in appendix A.

4 Results

	Average(s)	Maximum(s)	Minimum(s)	Size(KB)
Assembly	0.2199	0.5140	0.0850	1.624
CPP	0.6497	1.0240	0.5220	8.776
Python	4.3485	4.4040	4.3240	0.444
Java	26.3342	29.2470	23.0690	0.55

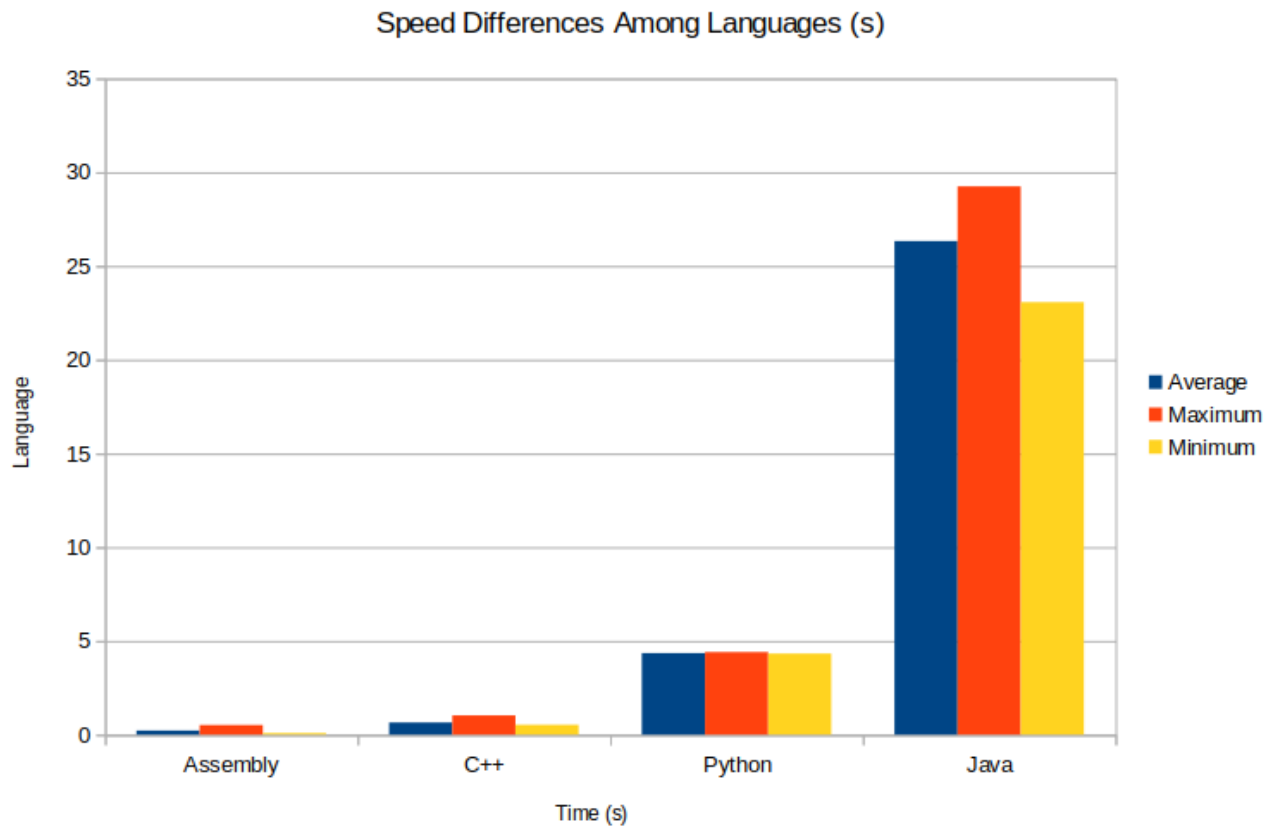


Figure 1: Graph showing execution speed of different languages

5 Discussion of Results

The assembly program was found to execute the fastest. This is due to being the lowest level language tested with. It assembles the instruction to binary (machine code). C++ is slightly slower. It is a compiled language, which is converted with a compiler into machine code. The size difference between is relatively small depending on the use of name-space standard library in C++ increasing the executable size. No default libraries need be implemented for assembly. The slower languages were Java and Python. These languages are byte-code interpreted languages. The CPU is required to call an interpreter which reads the code into byte code which is run by a virtual machine with the CPU. This process is slow. The executable sizes vary based on the libraries included and abstraction type with java written into a class by default when compiled with javac. The sizes however were quite small in this implementation from minimal library use.

6 Conclusion

A possible advantage in speed may be observed with the method of conversion to machine instructions. Assembly being the fastest in this case, compilation following closely behind and interpreter languages being the slowest. This however comes with a trade off of abstraction (design and application) and the complexity in code written and understood by the user.

A System Specifications

Summary

Computer	
Processor	Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
Memory	16280MB (1750MB used)
Machine Type	Notebook
Operating System	Ubuntu 20.04.1 LTS
User Name	liam (Liam Burgess)
Date/Time	Mon 17 Aug 2020 20:31:51

Figure 2: System specifications

Operating System

Version	
Kernel	Linux 5.4.0-42-generic (x86_64)
Version	#46-Ubuntu SMP Fri Jul 10 00:24:02 UTC 2020
C Library	GNU C Library / (Ubuntu GLIBC 2.31-0ubuntu9) 2.31
Distribution	Ubuntu 20.04.1 LTS

Figure 3: System specifications

Processor

Processors	
Package Information	
Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz	0 0:0 3500.00 MHz
Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz	1 0:1 3500.00 MHz
Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz	2 0:2 3500.00 MHz
Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz	3 0:3 3500.00 MHz

Figure 4: System specifications