University of Venda
Creating Future Leaders

FACULTY OF SCIENCE, ENGINEERING AND AGRICULTURE

DEPARTMENT OF MATHEMATICAL AND COMPUTATIONAL SCIENCE

INTRODUCTION TO COMPUTER SYSTEMS

COM 1122

ASSINGMENT 2

Group members Surname	Initials	Student Number
Hadzhi	FP	24029582
Nkanyane	NA	25059693
Ndwambi	T	25030044
Mutavhatsindi	W.B	25003806
Mmboneni	Z	25015755
Tshegameno	K.B	24035383
Mulaudzi	F.E	23021131
Manganye	V	24021003

Question 1

1.

1.1. The impact of CPU and Clock Speed on Computer Performance

The CPU is considered the brain of the computer. The CPU performs all types of data processing operations. It stores data, intermediate results, and instructions. It controls the operations of all parts of a computer. The clock speed measures the number of cycles the CPU executes per second, measured in GHz, one GHz is equivalent to one billion cycle per second, this means it will allow the facilities of multitasking.

a) The pursuit of the CPU performance is crucial for unlocking a computer's full potential. The CPU is considered the brain of the entire operation . The CPU helps the computer to process large amounts of information at a very high speed and produce accurate results.

b) The Clock Speed can be seen as the key factor in determine the CPU performance and how it is affects the computer's performance. In general, a higher clock speed means a faster CPU. However, many other factors come into play. A CPU processes many instructions from different programs per second, the clock speed measures the number of cycles a CPU executes per second, measured in GHz which facilitates multitasking and fast loads times.

1.2. The Role of Ram in Multitasking and System Performance

RAM is a form of temporary storage that holds data and programs while the computer is actively using them. When the user opens applications like a web browser, word processor, or music player, the system loads those programs into the **RAM** to allow quick access and smooth performance.

Role of Having More RAM

Having more RAM allows a computer to store more active data at once, which improves multitasking in many ways like:

a) Multiple applications can run simultaneously without slowing down.

b) The system can switch between programs more quickly

c) Large files and memory- intensive applications

For example, a computer with 8GB of RAM can web browers with multiple tabs, a word processor, and a music app smoothly. But a system with 4GB of

RAM may struggle and become sluggish when trying to handle all the programs at the same time.

Impact of Running Out Of RAM

If a system runs out of Ram, it uses a portion of the hard drive called virtual memory to compensate . However, virtual memory is much slower than RAM, programs may freeze, lag, or crash, the system may become unresponsive during switching or opening new apps..

Example:

a) **With enough RAM (8GB)**
A user opens Google Chrome with 10 tabs, Spotify, Microsoft word. Everything runs smoothly and switches between apps in an instant.

b) **With Low RAM (4GB)**
The same user opens the same programs. The system becomes slow, tabs reload constantly, apps freeze and take longer to switch between tasks.

1.3. HDDs and SSDs and its impact on Computer Speed

Comparing HDDS and SDDs

HDDs (Hard Disk Drive) uses mechanical parts- spinning disks and a moving read/write head.

SDD (Solid State Drive) has no moving parts and stores data on flash memory, like a USB flash drive.

Impact on Computer Speed

Features	HDDs	SDDs
Boot Time	30-60 seconds	5-15 seconds
File Access Speed	Slower (mechanical delays)	Much faster (instant access)
Data transfer	80-60 MB/s	500 MB/s or more (SATATA), up to 3500+MB/s (NVMe SSDs)
Durability	Prone to damage (Moving parts)	More durable (no moving parts)

Why SSDS Are Faster

SSDs results in faster data access because:

- They don't rely on mechanical movements to find related data.
- Data is accessed electronically, which is much quicker.
- This leads to faster boot times, applications launches, file transfers, and overall responsiveness.

Example of SSDs and HDDs in Real-world

- **Laptop with HDD:** Takes 40 seconds to boot up, and opening large apps like photoshop can take 20-30 seconds. File transfer are slow, and the system may lag when multitasking.
- **Laptop with SSD:** Boots in 10 seconds, open photoshop in 5 seconds, and file transfer happens much faster. The system feels more responsive and smoother even under pressure.

2.

2.1. Software Optimization and System Slowdown Over Time

Software optimization refers to the process of designing and updating software to run efficiently using minimal system resources such as CPU, RAM, and storage. A well-optimized program loads quickly, uses fewer resources, and performs tasks without unnecessary background processes.

Why System Slow Down Over Time

Several software-related reasons can cause a system to become slower over time even if the hardware is still in good condition.

a) Too many background processes
b) Unnecessary startup programs
c) Software bloat
d) Poorly optimized updates
e) Fragmented files (on HDDs)

Well-Optimized vs Poorly optimized proams

Feature	**Well-Optimized**	**Poorly Optimized Program**
Resource Usage	Minimal Ram and CPU use	High RAM/CPU usage even when it idle
Startup Behaviour	Opens quickly and does, not slow boot time	Slow boot-up and stays in background
Updates	Improves performance r fix bugs	Add feature that cause leg or crash
Impact on System	Keeps System responsive	Causes freezing, stuttering, or crashes

Example:

a) **Well-Optimized:** Google Chrome with Extension disabled and hardware acceleration enabled runs smoothly without hogging RAM.
b) **Poorly Optimized:** A basic photo editor that uses 1 GB of RAM just to open crashes often, and caused lag I other apps.

2.2. Choosing the Right Programming Language for Gaming and AI: C++ vs Python

When selecting a programming language for a project , it is important to understand the needs of the application. For a project that requires intensive gaming graphics performance it is a top priority. To make an informed recommendation, it helps to understand the difference between high-level and low-level programming languages, and why specific languages like C++ and Python are preferred for different purposes.

High Level vs Low Level Languages

- **High Level languages** are easier for humans to read and write. They are abstracted from the hardware, meaning the programmer does not need

to manage memory or deal with the hardware specific commands directly. Examples include Python, Java and JavaScript.
- **Low level languages** like Assembly or C are close to machine code and give the programmer more direct control over hardware resources, such as memory and CPU operations. C++ is considered a mid-level language because it combines features of both high and low level languages. It allows for high performance coding while still offering some abstraction.

Why we use C++ for Gaming

C++ is widely used in the game development industry because it provides high performance, fine control over memory, the ability to interact with graphics hardware. It supports object-oriented programming, which is useful for organizing complex game code, and it integrates easily with powerful game engines like Unreal Engine.

In a gaming environment, performance is critical. Games must run in real-time, often rendering complex 3D graphics, handling fast-paced interactions. C++ allows developers to write efficient code that executes quickly and uses system resources effectively.

Why we use Python For AI

On the hand, python is commonly used for artificial intelligence (AI) and machine learning projects. Python is easy to write and read, and it has a massive collection of libraries such as TensorFlow, Py Torch, and Scikit-learn, which make developing AI systems much faster and more accessible.

While Python is slower in execution compared to C++< AI tasks are often offloaded to optimized libraries written in C/C++ under the hood. This allows developers to write Python code while still benefiting from fast performance during computation-heavy operations.

Comparison of Performance in Use Cases

- **Gaming example:** A 3D shooting game developed in C++ runs at 120 frames per second (FPS) with smooth graphics and minimal lag because the code is optimized for direct communication with graphics hardware. If the same game were written in Python, it might struggle to maintain 30 FPS due to python’s slower execution and lack of low-level graphics control.
- **AI example:** A machine learning model for facial recognition is prototyped in Python using a few lines of code with the help of

libraries like OpenCV and TensorFlow. Writing the same model in C++ would take much longer, be harder to debug, and would not significantly improve performance thanks to Python's use of C++ based backends.

In summary, C++ excels in performance-critical applications like gaming, while Python shines in rapid development and algorithm-heavy tasks like AI.

2.3. Comparing Python and JavaScript for Web Development

Web developers need to select the appropriate language according to application's objective, performance needs, and scalability. Python and JavaScript are both widely used languages in web development, yet they have different purposes and provide unique benefits and drawbacks depending on the context in which they are used.

Python vs JavaScript

Python is a high level, versatile language commonly employed for backend web development, particularly with frameworks such as Django and Flask. It is appreciated for its ease of reading and quick development potential.

Benefits of Python For Web Development

a) **Clear and Manageable Code:** Python's straightforward syntax enables programmers to create brief and understandable code, accelerating the development process.

b) **Results Framework:** Frameworks such as Django offer integrated functionalities for security, database handling, and URL management.

c) **Reduced Execution Speed:** Python is an interpreted language, which oftens results in slower performance compared to complied or just-in-time (JIT) complied languages such as JavaScript in various scenarios.

d) **Restricted Browser Functionality:** Python cannot be executed within browsers, requiring it to be used alongside JavaScript.

e) **Concurrency Constraints:** The Global Interpreter (GIL) in python may restrict performance in web applications with high concurrency.

JavaScript is mainly utilized for front-end development, yet it can also function on the back-end through Node.js, it is crucial for creating interactive, dynamic user interfaces in web browsers.

Benefits of using JavaScript for website Development

a) **Operates in the Browser:** JavaScript is the sole language that can natively function in browsers, making it crucial for client-side development.

b) **Full stack Development:** Node.js enables JavaScript to be utilized on the frontend and backend, facilitating a smoother development workflow.

c) **Extensive Ecosystem:** Abundant libraries and freameworks (such as React, Angular, Vue) facilitate the creation of intricate and scalable applications.

REFERENCES

Aho, A.V., Lam, M.S., Sethi, R. and Ullman, J.D., 2007. Compliers: Principles, Techniques , and Tools. 2nd ed. Boston: Pearson/Addison Wesley. Andrews, R., 2022. Why does my computer slow down over time? TechRadar. Available at:

https://www.techradar.com [Accessed 13 Apr. 2025]

Boehm, B.W., 1989. A spiral model of software development and enhancement. ACM SIGSOFT Software Engineering Notes, 114(4), pp.13-24.

Flanagan, D., 2020. JavaScript: The Definitive Guide. 7th ed. Sebastopol, CA: O'Reilly Media.

Gregory, J., 2018. Game Engine Architecture. 3rd ed. A K Peters/CR Press.

Grover, R., 2015. Data Storage and Management for Cloud Computing. CRC Press.

Lutz, M., 2013. Learning Python. 5th ed. Sebastopol, CA: O'Reilly Media.

Microsoft, 2023. Tips to improve PC performance in Windows . Available at:

<httpss://support.microsoft.com> [Accessed 13 Apr.2025]. Patterson, D.A. and Hennessy, J.l., 2017. Computer Organization and Design RISC-V Edition: The Hardware Software interface. Morgen Kaufmann.

Russell, S . and Norvig, P., 2021. Artificial intelligence: A modern Approach. 4th ed. Pearson. Sebesta, R.W., 2012. Concepts of Programming Languages. 10th ed. Addison - Wesley.

Sillberschatz, A., Galvin, P.B. and Gagne, G., 2018. Operating system Concepts. 10th ed Willey.

Sommerville, I., 2016. Software Engineering. 10th ed. Boston: Pearson.

Stallings, W., 2018. Computer Organization and Architecture: Designing for Performance. 10th ed. Pearson.

Stroustrup, W., 2013. The C++Programming Language. 4th ed.Addision-wesley.

Van Rossum, G. and Drake, F.L., 2009. Python 3 Reference Manual. CreateSpace