Computer Environments and systems

You are smarter than computers! Trust yourself..

Computer components

- Two types:
  - Hardware and software
- Hardware is anything that you can touch/feel
- Software is anything that you cannot feel/touch or is virtual

Hardware Components

- **CPU (Central Processing Unit)**
  - Acts as the brain of the computer, executing instructions.
  - Higher clock speed and more cores lead to faster processing, crucial for multitasking and demanding applications.
- **RAM (Random Access Memory):**
  - Provides temporary storage for data and instructions currently in use by the CPU.
  - More RAM allows for smoother multitasking and handling of large files or applications.
- **ROM (Read-Only Memory):**
  - Stores firmware and essential system instructions.
  - Remains intact even when the computer is powered off, ensuring essential startup processes.
- **Cache:**
  - Provides high-speed storage directly accessible by the CPU.
  - Faster access to frequently used data, enhancing overall system performance.

Hardware Components

- **Hard Drive (or Solid State Drive – SSD):**
  - Stores data persistently, including the operating system, software, and user files.
  - Higher capacity drives offer more storage space for multimedia files, programs, and documents.
- **Motherboard:**
  - Connects and coordinates communication between all internal components.
  - Compatibility with CPU, RAM, and expansion cards is crucial for system stability and performance.
- **Power Supply:**
  - Supplies electrical power to all components within the computer.
  - Sufficient wattage and stable voltage output are essential for reliable operation.
- **Video Card (Graphics Processing Unit - GPU):**
  - Handles rendering of graphics and visuals.
  - Higher performance GPUs enable smooth gaming, video editing, and graphical workloads.
- **Sound Card:**
  - Processes audio signals for playback and recording.
  - Higher quality sound cards deliver clearer audio and support for advanced audio features.

Group Activity

- In your group, play the following game. The group with the lowest time and the most correct answer wins:

https://wordwall.net/resource/10827661/computing/parts-of-a-computer

- Peripheral Devices:
  - These are not essential to the functioning of a computer but enhances its functionality. It can be both output and input device!
  - E.G.- printers, monitor, scanner, keyboard, mouse, speaker, USB drives called?

Individual Activity - Computer Hardware

- **More to read: https://edu.gcfglobal.org/en/computerbasics/inside-a-computer/1/

- Make a summary in your own words that explains the function of each major component of a computer system based on what you read and watched.
- Make a list of 10 things you think are the most important things to understand about computer hardware for somebody who doesn't understand anything about computers.
- Submit it in google classroom

Processors!

- To be continued!

Things get a bit more complex!-- Software

- Software refers to a collection of instructions, programs, and data that enable computers to perform specific tasks or functions. It encompasses everything from operating systems to applications and utilities.
- Types of software:
  - Operating Systems
  - Application Software
  - Utility Software
  - Programing Software
  - System Software

****Your task- In group, give ateast two examples for each types of software**

Types of Software

- **Operating Systems (OS)**
  - Controls hardware resources and provides a platform for running applications.
  - Examples:
- **Application Software:**
  - Designed for specific tasks or user needs, such as word processing, graphic design, web browsing, and multimedia playback.
  - Examples:
- **Utility Software**
  - Aids in managing, maintaining, and optimizing computer systems, ensuring security, performance, and data integrity.
  - Examples:

Types of Software

- **Programming Software:**
  - Provides tools for developers to write, test, and debug software applications and scripts.
  - Examples:
- **Systems Software:**
  - Facilitates communication between hardware components and the operating system, ensuring proper functionality.
  - Examples:

Types of Software

- **Operating Systems (OS)**
  - Controls hardware resources and provides a platform for running applications.
  - Examples: Windows, macOS, Linux, Android, iOS
- **Application Software:**
  - Designed for specific tasks or user needs, such as word processing, graphic design, web browsing, and multimedia playback.
  - Examples: Microsoft Office Suite (Word, Excel, PowerPoint), Adobe Photoshop, Google Chrome, Spotify
- **Utility Software**
  - Aids in managing, maintaining, and optimizing computer systems, ensuring security, performance, and data integrity.
  - Examples: Antivirus software (Norton, McAfee), Disk Cleanup tools, Backup software

Types of Software

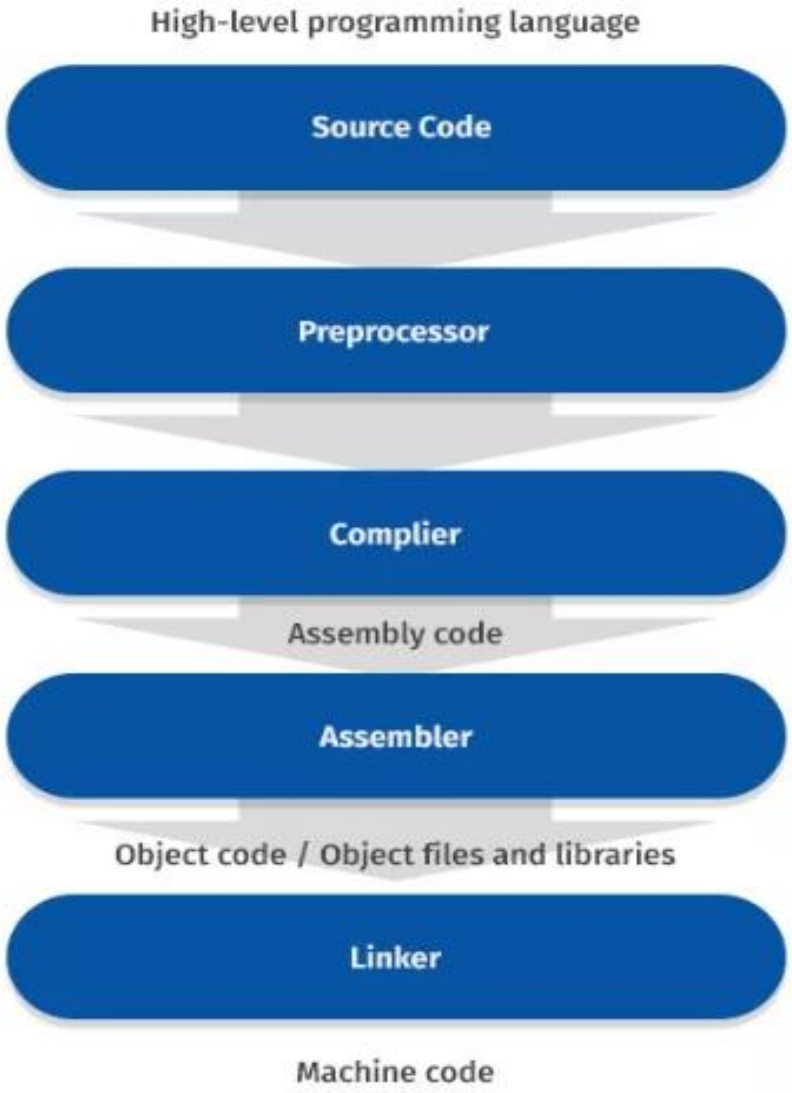
- **Programming Software:**
  - Provides tools for developers to write, test, and debug software applications and scripts.
  - Examples: Integrated Development Environments (IDEs) like Visual Studio, PyCharm, Eclipse
- **Systems Software:**
  - Facilitates communication between hardware components and the operating system, ensuring proper functionality.
  - Examples: Device drivers, Firmware

Why do we need software?

- Enables Functionality: Without software, computers would be inert machines incapable of performing tasks.
- Enhances Productivity: Applications streamline processes and automate tasks, boosting efficiency and productivity.
- Facilitates Communication: Software enables communication through email clients, messaging apps, and social media platforms.
- Supports Innovation: Constant development and improvement of software drive technological advancements and innovation across industries.

How does the computer understand what you type or give as input?

- A high-level language (HLL) is a programming language, such as C, python, java, PHP, C# etc. that allows a programmer to develop programmes that are independent of the type of machine they are running on
- Computer does not understand high level language---Lol!
- Computers only understand machine code which is in binary either 0s or 1s
- Before next class, read the following:
  - https://optima-systems.co.uk/how-does-a-computer-understand-a-programming-language/#:~:text=Computers%20only%20understand%20machine%20code,binary%20code%200s%20and%201s.



High Level Vs Low Level

- Programming languages can be categorized based on their abstraction level, with high-level and low-level languages representing two ends of the spectrum. Understanding their differences is crucial for choosing the right tool for specific programming tasks.

High Level Vs Low Level

Criteria	High Level Programming/Source Code	Low Level Programming/Machine Code
Abstraction Level	High-level languages abstract away hardware details, focusing on solving problems.	Low-level languages offer minimal abstraction, providing direct control over hardware resources.
Ease of Use	High-level languages are more user-friendly, with syntax closer to natural language.	Low-level languages require a deeper understanding of hardware and typically involve more complex syntax.
Performance	High-level languages prioritize ease of development over raw performance, although modern high-level languages are often optimized for speed.	Low-level languages offer greater control over performance optimization, resulting in potentially faster and more efficient code.
Portability	High-level languages are more portable, as they are designed to run on different platforms with minimal modifications.	Low-level languages may require significant adaptation to run on different hardware architectures or operating systems.
Example	Python, Java, C#, JavaScript, Ruby etc.	Assembly language, Machine code.