Syllabus

Unit 1:

Introduction to Computer Fundamentals: Introduction to Computer, Objectives, Hardware and Software, Block Diagram of the Computer, Functions of the different Units, Applications of Computers Representation of data and information, Computer Languages, Machine language, Assembly Language, High level Language, Number System and Conversion, Classification and History of Computers, Introduction to Free and Open Source Software, Computer Virus, Use of Antivirus software.

What is Computer?

A computer is a programmable electronic device that accepts raw data as input and processes it with a set of instructions (a program) to produce the result as output. It renders output just after performing mathematical and logical operations and can save the output for future use. It can process numerical as well as non-numerical calculations. The term "computer" is derived from the Latin word "computare" which means to calculate.

A computer is designed to execute applications and provides a variety of solutions through integrated hardware and software components. It works with the help of programs and represents the decimal numbers through a string of binary digits. It also has a memory that stores the data, programs, and result of processing. The components of a computer such as machinery that includes wires, transistors, circuits, hard disk are called hardware. Whereas, the programs and data are called software.

It is believed that the Analytical Engine was the first computer which was invented by Charles Babbage in 1837. It used punch cards as read-only memory. Charles Babbage is also known as the father of the computer.

The basic parts without which a computer cannot work are as follows:

- **Processor:** It executes instructions from software and hardware.
- **Memory:** It is the primary memory for data transfer between the CPU and storage.
- **Motherboard:** It is the part that connects all other parts or components of a computer.
- **Storage Device:** It permanently stores the data, e.g., hard drive.
- **Input Device:** It allows you to communicate with the computer or to input data, e.g., a keyboard.
- **Output Device:** It enables you to see the output, e.g., monitor.

Computers are divided into different types based on different criteria. Based on the size, a computer can be divided into five types:

1. Micro Computer
2. Mini Computer
3. Mainframe Computer
4. Super Computer
5. Workstations

---

1. Micro Computer:

It is a single-user computer which has less speed and storage capacity than the other types. It uses a microprocessor as a CPU. The first microcomputer was built with 8-bit microprocessor chips. The common examples of microcomputers include laptops, desktop computers, personal digital assistant (PDA), tablets, and smartphones. Microcomputers are generally designed and developed for general usage like browsing, searching for information, internet, MS Office, social media, etc.

2. Mini Computer:

Mini-computers are also known as "Midrange Computers." They are not designed for a single. They are multi-user computers designed to support multiple users simultaneously. So, they are generally used by small businesses and firms. Individual departments of a company use these computers for specific purposes. For example, the admission department of a University can use a Mini-computer for monitoring the admission process.

3. Mainframe Computer:

It is also a multi-user computer capable of supporting thousands of users simultaneously. They are used by large firms and government organizations to run their business operations as they can store and process large amounts of data. For example, Banks, universities, and insurance companies use mainframe computers to store the data of their customers, students, and policyholders, respectively.

4. Super Computer:

Super-computers are the fastest and most expensive computers among all types of computers. They have huge storage capacities and computing speeds and thus can perform millions of instructions per second. The super-computers are task-specific and thus used for specialized applications such as large-scale numerical problems in scientific and engineering

disciplines including applications in electronics, petroleum engineering, weather forecasting, medicine, space research and more. For example, NASA uses supercomputers for launching space satellites and monitoring and controlling them for space exploration.

5. Work stations:

It is a single-user computer. Although it is like a personal computer, it has a more powerful microprocessor and a higher-quality monitor than a microcomputer. In terms of storage capacity and speed, it comes between a personal computer and minicomputer. Work stations are generally used for specialized applications such as desktop publishing, software development, and engineering designs.

Benefits of Using a Computer:

- **Increases your productivity:** A computer increases your productivity. For example, after having a basic understanding of a word processor, you can create, edit, store, and print the documents easily and quickly.
- **Connects to the Internet:** It connects you to the internet that allows you to send emails, browse content, gain information, use social media platforms, and more. By connecting to the internet, you can also connect to your long-distance friends and family members.
- **Storage:** A computer allows you to store a large amount of information, e.g., you can store your projects, ebooks, documents, movies, pictures, songs, and more.
- **Organized Data and Information:** It not only allows you to store data but also enables you to organize your data. For example, you can create different folders to store different data and information and thus can search for information easily and quickly.
- **Improves your abilities:** It helps write good English if you are not good at spelling and grammar. Similarly, if you are not good at math, and don't have a great memory, you can use a computer to perform calculations and store the results.
- **Assist the physically challenged:** It can be used to help the physically challenged, e.g., Stephen Hawking, who was not able to speak used computer to speak. It also can be used to help blind people by installing special software to read what is on the screen.
- **Keeps you entertained:** You can use the computer to listen to songs, watch movies, play games and more.

The computer has become a part of our life. There are plenty of things that we do in a day are dependent on a computer. Some of the common examples are as follows:

1. **ATM:** While withdrawing cash from an ATM, you are using a computer that enables the ATM to take instructions and dispense cash accordingly.
2. **Digital currency:** A computer keeps a record of your transactions and balance in your account and the money deposited in your account in a bank is stored as a digital record or digital currency.
3. **Trading:** Stock markets use computers for day to day trading. There are many advanced algorithms based on computers that handle trading without involving humans.
4. **Smartphone:** The smartphone that we use throughout the day for calling, texting, browsing is itself a computer.
5. **VoIP:** All voice over IP communication (VoIP) is handled and done by computers.

History of Computers

The first counting device was used by the primitive people. They used sticks, stones and bones as counting tools. As human mind and technology improved with time more computing devices were developed. Some of the popular computing devices starting with the first to recent ones are described below;

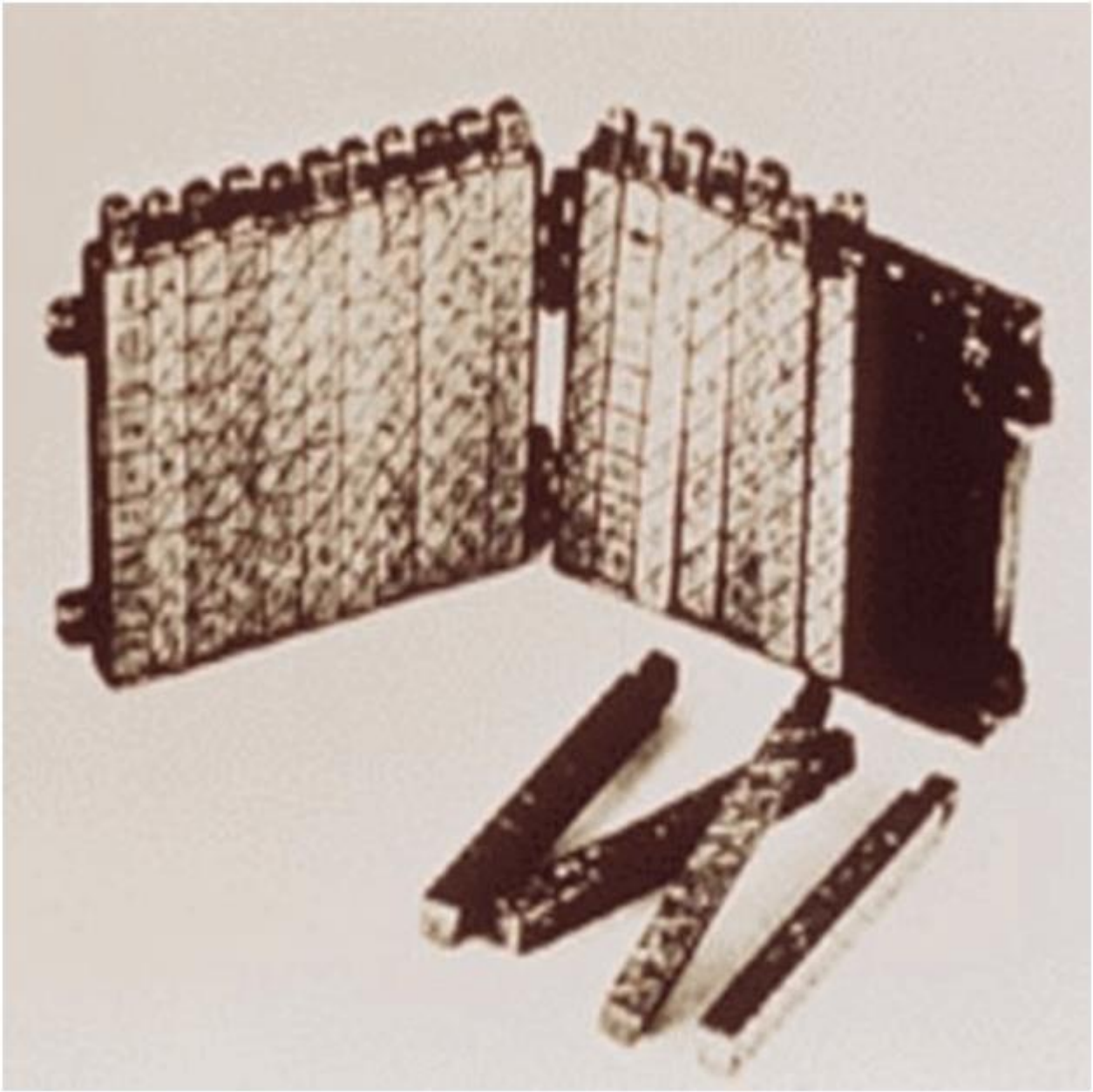
Abacus

The history of computer begins with the birth of abacus which is believed to be the first computer. It is said that Chinese invented Abacus around 4,000 years ago.

It was a wooden rack which has metal rods with beads mounted on them. The beads were moved by the abacus operator according to some rules to perform arithmetic calculations. Abacus is still used in some countries like China, Russia and Japan. An image of this tool is shown below;

Napier's Bones

It was a manually-operated calculating device which was invented by John Napier (1550-1617) of Merchiston. In this calculating tool, he used 9 different ivory strips or bones marked with numbers to multiply and divide. So, the tool became known as "Napier's Bones. It was also the first machine to use the decimal point.



Pascaline

Pascaline is also known as Arithmetic Machine or Adding Machine. It was invented between 1642 and 1644 by a French mathematician-philosopher Biaise Pascal. It is believed that it was the first mechanical and automatic calculator.

Pascal invented this machine to help his father, a tax accountant. It could only perform addition and subtraction. It was a wooden box with a series of gears and wheels. When a wheel is rotated one revolution, it rotates the neighboring wheel. A series of windows is given on the top of the wheels to read the totals. An image of this tool is shown below;

Stepped Reckoner or Leibnitz wheel

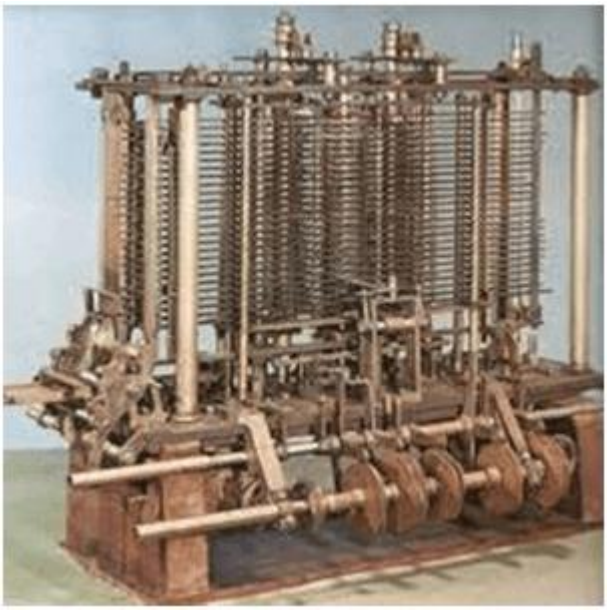
It was developed by a German mathematician-philosopher Gottfried Wilhelm Leibnitz in 1673. He improved Pascal's invention to develop this machine. It was a digital mechanical calculator which was called the stepped reckoner as instead of gears it was made of fluted drums. See the following image;

Difference Engine

In the early 1820s, it was designed by Charles Babbage who is known as "Father of Modern Computer". It was a mechanical computer which could perform simple calculations. It was a steam driven calculating machine designed to solve tables of numbers like logarithm tables.

Analytical Engine

This calculating machine was also developed by Charles Babbage in 1830. It was a mechanical computer that used punch-cards as input. It was capable of solving any mathematical problem and storing information as a permanent memory.

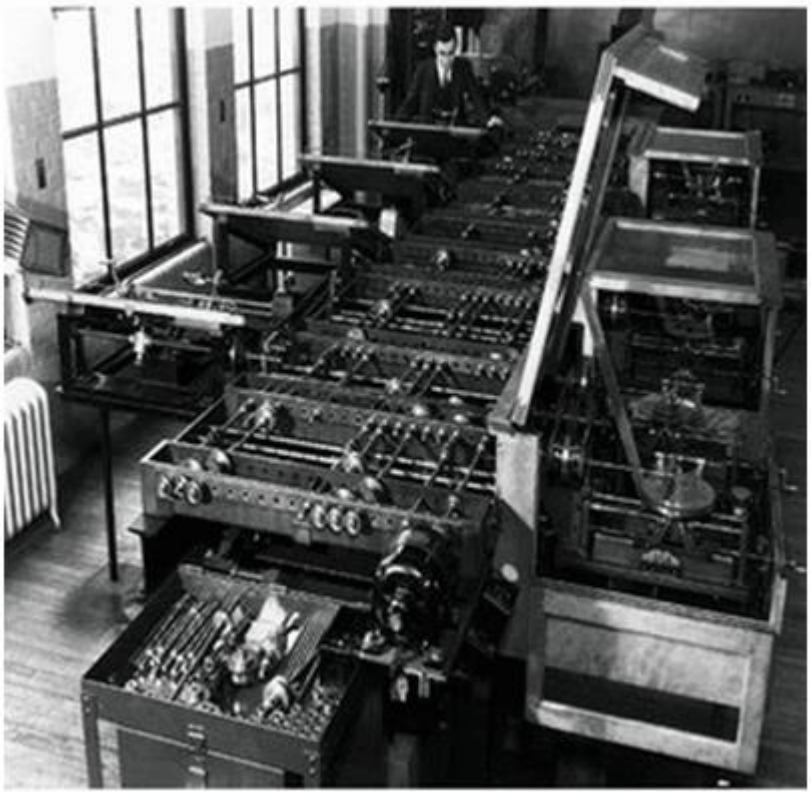


Tabulating Machine

It was invented in 1890, by Herman Hollerith, an American statistician. It was a mechanical tabulator based on punch cards. It could tabulate statistics and record or sort data or information. This machine was used in the 1890 U.S. Census. Hollerith also started the Hollerith?s Tabulating Machine Company which later became International Business Machine (IBM) in 1924.

Differential Analyzer

It was the first electronic computer introduced in the United States in 1930. It was an analog device invented by Vannevar Bush. This machine has vacuum tubes to switch electrical signals to perform calculations. It could do 25 calculations in few minutes.



Mark I

The next major changes in the history of computer began in 1937 when Howard Aiken planned to develop a machine that could perform calculations involving large numbers. In 1944, Mark I computer was built as a partnership between IBM and Harvard. It was the first programmable digital computer.

Generations of Computers

A generation of computers refers to the specific improvements in computer technology with time. In 1946, electronic pathways called circuits were developed to perform the counting. It replaced the gears and other mechanical parts used for counting in previous computing machines.

In each new generation, the circuits became smaller and more advanced than the previous generation circuits. The miniaturization helped increase the speed, memory and power of computers. There are five generations of computers which are described below;

First Generation Computers

The first generation (1946-1959) computers were slow, huge and expensive. In these computers, vacuum tubes were used as the basic components of CPU and memory. These computers were mainly depended on batch operating system and punch cards. Magnetic tape and paper tape were used as output and input devices in this generation;

Some of the popular first generation computers are;

- **ENIAC** (Electronic Numerical Integrator and Computer)
- **EDVAC** (Electronic Discrete Variable Automatic Computer)
- **UNIVACI**(Universal Automatic Computer)
- **IBM-701**
- **IBM-650**

Second Generation Computers

The second generation (1959-1965) was the era of the transistor computers. These computers used transistors which were cheap, compact and consuming less power; it made transistor computers faster than the first generation computers.

In this generation, magnetic cores were used as the primary memory and magnetic disc and tapes were used as the secondary storage. Assembly language and programming languages like COBOL and FORTRAN, and Batch processing and multiprogramming operating systems were used in these computers.

Some of the popular second generation computers are;

- **IBM 1620**

- **IBM 7094**
- **CDC 1604**
- **CDC 3600**
- **UNIVAC 1108**

Third Generation Computers

The third generation computers used integrated circuits (ICs) instead of transistors. A single IC can pack huge number of transistors which increased the power of a computer and reduced the cost. The computers also became more reliable, efficient and smaller in size. These generation computers used remote processing, time-sharing, multi programming as operating system. Also, the high-level programming languages like FORTRON-II TO IV, COBOL, PASCAL PL/1, ALGOL-68 were used in this generation.

Some of the popular third generation computers are;

- **IBM-360 series**
- **Honeywell-6000 series**
- **PDP(Personal Data Processor)**
- **IBM-370/168**
- **TDC-316**

Fourth Generation Computers

The fourth generation (1971-1980) computers used very large scale integrated (VLSI) circuits; a chip containing millions of transistors and other circuit elements. These chips made this generation computers more compact, powerful, fast and affordable. These generation computers used real time, time sharing and distributed operating system. The programming languages like C, C++, DBASE were also used in this generation.

Some of the popular fourth generation computers are;

- **DEC 10**
- **STAR 1000**
- **PDP 11**
- **CRAY-1(Super Computer)**
- **CRAY-X-MP(Super Computer)**

Fifth Generation Computers

In fifth generation (1980-till date) computers, the VLSI technology was replaced with ULSI (Ultra Large Scale Integration). It made possible the production of microprocessor chips with ten million electronic components. This generation computers used parallel processing hardware and AI (Artificial Intelligence) software. The programming languages used in this generation were C, C++, Java, .Net, etc.

Some of the popular fifth generation computers are;

- **Desktop**
- **Laptop**
- **NoteBook**
- **UltraBook**
- **ChromeBook**

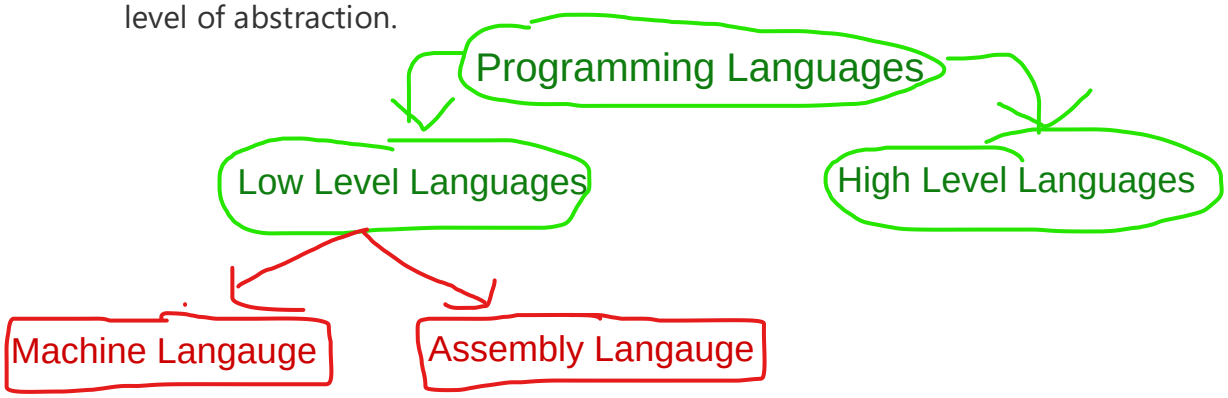
What is a programming language?

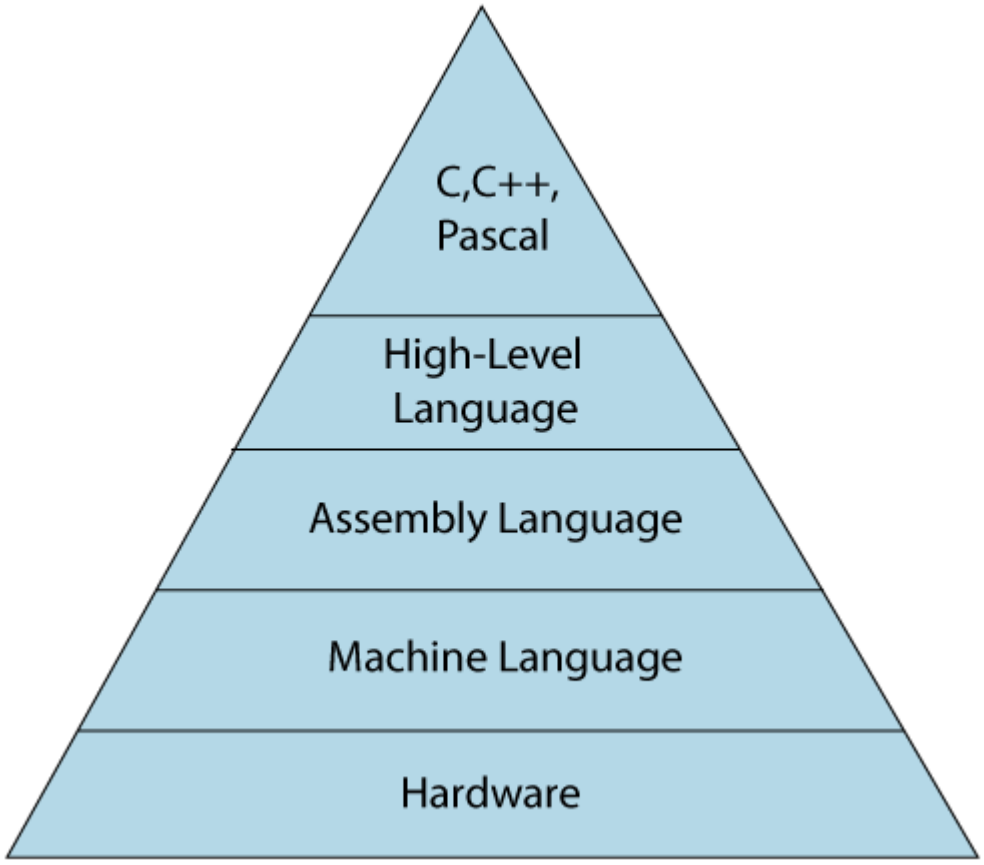
A programming language defines a set of instructions that are compiled together to perform a specific task by the CPU (Central Processing Unit). The programming language mainly refers to high-level languages such as C, C++, Pascal, Ada, COBOL, etc.

Each programming language contains a unique set of keywords and syntax, which are used to create a set of instructions. Thousands of programming languages have been developed till now, but each language has its specific purpose. These languages vary in the level of abstraction they provide from the hardware. Some programming languages provide less or no abstraction while some provide higher abstraction. Based on the levels of abstraction, they can be classified into two categories:

- Low-level language
- High-level language

The image which is given below describes the abstraction level from hardware. As we can observe from the below image that the machine language provides no abstraction, assembly language provides less abstraction whereas high-level language provides a higher level of abstraction.





Low-level language

The low-level language is a programming language that provides no abstraction from the hardware, and it is represented in 0 or 1 forms, which are the machine instructions. The languages that come under this category are the Machine level language and Assembly language.

Machine-level language

The machine-level language is a language that consists of a set of instructions that are in the binary form 0 or 1. As we know that computers can understand only machine instructions, which are in binary digits, i.e., 0 and 1, so the instructions given to the computer can be only in binary codes. Creating a program in a machine-level language is a very difficult task as it is not easy for the programmers to write the program in machine instructions. It is error-prone as it is not easy to understand, and its maintenance is also very high. A machine-level language is not portable as each computer has its machine instructions, so if we write a program in one computer will no longer be valid in another computer.

The different processor architectures use different machine codes, for example, a PowerPC processor contains RISC architecture, which requires different code than intel x86 processor, which has a CISC architecture.

Assembly Language

The assembly language contains some human-readable commands such as mov, add, sub, etc. The problems which we were facing in machine-level language are reduced to some extent by using an extended form of machine-level language known as assembly language. Since assembly language instructions are written in English words like mov, add, sub, so it is easier to write and understand.

As we know that computers can only understand the machine-level instructions, so we require a translator that converts the assembly code into machine code. The translator used for translating the code is known as an assembler.

The assembly language code is not portable because the data is stored in computer registers, and the computer has to know the different sets of registers.

The assembly code is not faster than machine code because the assembly language comes above the machine language in the hierarchy, so it means that assembly language has some abstraction from the hardware while machine language has zero abstraction.

Differences between Machine-Level language and Assembly language

The following are the differences between machine-level language and assembly language:

Machine-level language	Assembly language
The machine-level language comes at the lowest level in the hierarchy, so it has zero abstraction level from the hardware.	The assembly language comes above the machine language means that it has less abstraction level from the hardware.
It cannot be easily understood by humans.	It is easy to read, write, and maintain.
The machine-level language is written in binary digits, i.e., 0 and 1.	The assembly language is written in simple English language, so it is easily understandable by the users.
It does not require any translator as the machine code is directly executed by the computer.	In assembly language, the assembler is used to convert the assembly code into machine code.
It is a first-generation programming language.	It is a second-generation programming language.

High-Level Language

The high-level language is a programming language that allows a programmer to write the programs which are independent of a particular type of computer. The high-level languages are considered as high-level because they are closer to human languages than machine-level languages.

When writing a program in a high-level language, then the whole attention needs to be paid to the logic of the problem.

A compiler is required to translate a high-level language into a low-level language.

Advantages of a high-level language

- The high-level language is easy to read, write, and maintain as it is written in English like words.
- The high-level languages are designed to overcome the limitation of low-level language, i.e., portability. The high-level language is portable; i.e., these languages are machine-independent.

Differences between Low-Level language and High-Level language

The following are the differences between low-level language and high-level language:

Low-level language	High-level language
It is a machine-friendly language, i.e., the computer understands the machine language, which is represented in 0 or 1.	It is a user-friendly language as this language is written in simple English words, which can be easily understood by humans.
The low-level language takes more time to execute.	It executes at a faster pace.
It requires the assembler to convert the assembly code into machine code.	It requires the compiler to convert the high-level language instructions into machine code.
The machine code cannot run on all machines, so it is not a portable language.	The high-level code can run all the platforms, so it is a portable language.
It is memory efficient.	It is less memory efficient.
Debugging and maintenance are not easier in a	Debugging and maintenance are easier in a high-

low-level language.	level language.

Computer Virus

Computer viruses are unwanted software programs or pieces of code that interfere with the functioning of the computer. They spread through contaminated files, data, and insecure networks. Once it enters your system, it can replicate to produce copies of itself to spread from one program to another program and from one infected computer to another computer. So, we can say that it is a self-replicating computer program that interferes with the functioning of the computer by infecting files, data, programs, etc.

There are many antiviruses, which are programs that can help you protect your machine from viruses. It scans your system and cleans the viruses detected during the scan. Some of the popular antiviruses include Avast, Quickheal, McAfee, Kaspersky, etc.

Types of Computer Virus:

Overwrite Virus:

It is the simplest computer virus that overwrites the code of the host computer system's file with its own malicious code. The content of the infected file is replaced partially or completely without changing the size of the file. Thus, it destroys the original program code by overwriting it with its defective code. The infected files must be deleted or replaced with a new copy as this virus cannot be removed or disinfected.

Append Virus:

As the name suggests, this virus appends its malicious code to the end of the host program's file. After that, it alters the file's header in a way that the file's header is redirected to the start of the malicious code of the append virus. Thus, this code is executed each time the program runs. However, it does not destroy the host program; rather, it modifies it in a way that it holds the virus code and enables the code to run itself.

Macro Virus

Macro virus alters or infects the macros of a document or data file. It is embedded as a macro in a document and adds its codes to the macros of the document. The virus spreads when infected documents or data files are opened in other computers.

It also spreads through software programs, which execute macros such as Ms Word, Ms Excel. Each time a document is opened using these programs, other related documents will also get infected.

The first macro virus, which was named concept, spread through emails with attached Ms Word documents. It infected MsWord 6.0 and Ms Word 95 documents, which were saved using Save As option. Fortunately, it did not cause any harm, except for displaying a message on the screen.

Boot Virus

Boot virus or boot sector virus alters the boot sector program stored in the hard disk or any other storage device such as floppy disks. It replaces the boot sector program with its own malicious version. It infects the computer only when it is used to boot up the computer. If it enters after the boot-up process, it will not infect the computer. For example, if someone forgets to remove the infected floppy disk when the pc is turned off and then turns on this pc, it runs the infected boot sector program during the booting process.

Usually, it enters into your system through corrupt media files, infected storage devices, and insecure computer networks. The spread of this virus is very rare these days due to the decline in the use of floppy disk and use of boot-sector safeguards in the present-day operating systems.

Resident Virus

The resident virus stays permanently in the primary memory (RAM) of the computer. When you start the computer, it becomes active and corrupts the files and programs running on the computer.

Non-resident Virus:

Unlike the resident virus, the non-resident virus does not reside in the memory of a computer. So, it is not executed from the computer's memory. For example, executable viruses.

Multipartite Virus

Multipartite virus spreads and infects in multiple ways. It infects both the boot sector and the executable files stored on the hard drive simultaneously. When you turn on a computer,

the boot sector virus is triggered as it latches on to the hard drive, which has the data for starting up the computer. Once it is triggered, the program files also get infected.

File Infector Virus

It is one of the commonly found computer viruses. It mainly infects the executable files; the files with .com or .exe extensions. The virus becomes active when the infected file is executed. The active virus overwrites the file partially or completely. Thus it may destroy the original file partially or completely.

Computer Worm

Computer worm is similar to a virus but is technically different from the virus. It can replicate and spread like a virus, but unlike viruses, it does not need a host program to spread. Being able to self-replicate it can produce multiple copies of itself. It spreads through networks such as an email sent to an infected email id can infect your system with a computer worm.

Trojan Horse

Trojan horse is a malware like a virus or a worm, but it is technically different from both. It can't replicate like virus and worm. Trojan horse hides itself in a program. Once you install any such program, the trojan horse enters into your computer. It can provide unauthorized access to your computer, send your files to other computers, and may delete files or can make other unwanted changes in your computer.

Cavity virus:

It is also known as a spacefiller virus. As the name suggests, this virus tends to install itself by occupying the empty sections of a file. It is not easy to detect this virus as it fills the empty spaces without changing the size of the file.

CMOS Virus:

It infects the CMOS, which stands for complementary metal-oxide semiconductor and is a memory chip that contains the system configuration. This virus can erase or reset the system configuration.

Companion Virus:

It resides itself in a file whose name is similar to another program file, which is executed normally. When the program file is executed, the virus gets activated and performs malicious steps such as deleting the files on your computer hard drive. Globe virus is a first known companion virus, which was found in 1992.

Encrypted Virus:

It encrypts its payload to make its detection more difficult. It comprises two parts: an encrypted virus body and a decryptor, which decrypts the virus when it is executed. After decryption, the virus can execute itself in order to replicate and become a resident. Furthermore, it is different from cryptolocker, which is a computer virus that encrypts the hard drive data and holds it for ransom.

Executable Virus:

It is a non-resident computer virus, which resides in an executable file. Whenever the infected file is executed, it infects the other files.

Polymorphic Virus:

It creates its thousands of copies itself; in each copy, it changes the sequence and byte values to evade detection by antivirus software. Even the best antiviruses may not be able to detect this virus. Polymorphic viruses affect data types and functions and generally spread through spam, infected sites, and while using other malware.

Rabbit Virus:

It is also known as wabbit, a fork bomb. It is capable of creating new processes, and each of the new process further creates new processes. This process continues until this virus utilizes all the available resources in the system and system falls short of resources. It may cause the target system to slow down and crash. For example, it is like an Infinite loop that repeatedly creates processes that consume lots of CPU cycles and operating system resources.

Stealth Virus:

It is a hidden computer virus, which specifically attacks operating system processes. It usually hides itself in partitions, files or boot sectors and is capable of going unnoticed during antivirus or anti-malware scans, i.e., it can avoid detection intentionally.

Symptoms of a Computer Virus:

There are many warning signs or symptoms which show that a computer is infected with a virus, some of which are as follows:

- **Slow computer performance:** The machine may work slowly, e.g., it will take more time to open or shut down the computer or while opening a file, document, computer application, etc. The operating system and internet speed may get slow.
- **Frequent pop-ups:** A virus may cause unusual frequent pop-ups on your window.

- **Hard Drive issue:** The hard drive may exhibit unusual high activity even when it is not in use. It may cause unwanted changes to your hard drive and may freeze or crash this device.
- **Frequent crashes:** One may experience frequent sudden system crashes while playing games, watching videos, or doing some other work using the infected system. A blue screen appears when it crashes.
- **Unknown programs:** Unwanted programs may open or start automatically when you start your computer. You can see these programs in your computer's list of active applications. Sometimes, the window shuts down unexpectedly without any reason.
- **Unusual activities:** Your machine may perform differently, such as you may not be able to log into your accounts, to delete the corrupt files, and Blue Screen of Death (BSOD) may appear frequently, and more. Furthermore, the hardware, software, or OS may start malfunctioning leading to crashing the system abruptly.
- **Impaired security solutions:** Sometimes, security measures taken by you, such as antivirus may not work smoothly due to virus attack on your computer.
- **Network issue:** Sometimes, you experience high network activity even if you are not connected to the internet and vice versa.
- **Unnecessary advertisement:** We often see advertisements while browsing, but if you see them even when you are not browsing, it may indicate a virus on your computer.
- **Display problems:** You may experience different colors in your display if your computer is affected by a virus.
- **Affected Applications:** Some viruses are developed to affect specific applications. Consequently, some applications may not work on your computer if it is infected.
- **Blocked by Antivirus Sites:** An antivirus site may deny access to a computer that is infected by a virus.
- **Dialog Boxes:** Many dialog boxes keep appearing suddenly on your screen.
- **Printer Issues:** A printer attached to an infected computer may print documents without getting any command or in an inappropriate manner.
- **Changed Homepage:** Your home page may get changed without any effort from your side. For example, you may see a new toolbar on your screen, and you may be redirected to a different web address instead of the page visited by you initially.
- **Strange messages:** One may see strange messages on a computer screen such as error messages, e.g., "cannot rename "folder" as a folder already exists with this name"

Number Systems

The language we use to communicate with each other is comprised of words and characters. We understand numbers, characters and words. But this type of data is not suitable for computers. Computers only understand the numbers.

So, when we enter data, the data is converted into electronic pulse. Each pulse is identified as code and the code is converted into numeric format by ASCII. It gives each number, character and symbol a numeric value (number) that a computer understands. So to understand the language of computers, one must be familiar with the number systems.

The Number Systems used in computers are:

- Binary number system
- Octal number system
- Decimal number system
- Hexadecimal number system

Binary number system

It has only two digits '0' and '1' so its base is 2. Accordingly, In this number system, there are only two types of electronic pulses; absence of electronic pulse which represents '0'and presence of electronic pulse which represents '1'. Each digit is called a bit. A group of four bits (1101) is called a nibble and group of eight bits (11001010) is called a byte. The position of each digit in a binary number represents a specific power of the base (2) of the number system.

Octal number system

It has eight digits (0, 1, 2, 3, 4, 5, 6, 7) so its base is 8. Each digit in an octal number represents a specific power of its base (8). As there are only eight digits, three bits (23=8) of binary number system can convert any octal number into binary number. This number system is also used to shorten long binary numbers. The three binary digits can be represented with a single octal digit.

Decimal number system

This number system has ten digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) so its base is 10. In this number system, the maximum value of a digit is 9 and the minimum value of a digit is 0. The position of each digit in decimal number represents a specific power of the base (10) of the

number system. This number system is widely used in our day to day life. It can represent any numeric value.

Hexadecimal number system

This number system has 16 digits that ranges from 0 to 9 and A to F. So, its base is 16. The A to F alphabets represent 10 to 15 decimal numbers. The position of each digit in a hexadecimal number represents a specific power of base (16) of the number system. As there are only sixteen digits, four bits (24=16) of binary number system can convert any hexadecimal number into binary number. It is also known as alphanumeric number system as it uses both numeric digits and alphabets.

What does Open-source mean?
The term **Open-source** is closely related to Open-source software (OSS). Open-source software is a type of computer software that is released under a license, but the source code is made available to all the users. The copyright holders of such software allow the users to use it and do some valuable modifications in its source code to add some new features, to improve the existing features, and to fix bugs if there are any. Because of this reason only Open-source software is mostly developed collaboratively.
Some famous examples of Open-source products are :

- **Operating systems –**
  Android, Ubuntu, Linux
- **Internet browsers –**
  Mozilla Firefox, Chromium
- **Integrated Development Environment (IDEs) –**
  Vs code (Visual Studio Code), Android Studio, PyCharm, Xcode

Open-source community and Contributions :
The **open-source community** is a worldwide community of programmers and software developers who are continuously working on various open-source projects to make our lives better. This community is self-governing and self-organizing, there are no executives to take the decisions solely. This community plays a very crucial role in the sustainability of various open-source organizations.
The contributions made in any open-source project which improves its usability are called **open-source contributions**. These contributions can be of any form not only some software codes like we can work on improving its **documentation**, improving its **UI/UX (user interface and design)**, organize meetups, or find new collaborators.
Benefits of Open-source contributions :

- We code for real-world open-source projects.
- It refines our existing knowledge of programming and also helps us to learn new skills.
- Many open-source projects offer mentorship programs to guide and help us through our first few contributions.
- We need not develop the whole thing from scratch, we just have to fork our favorite projects and start experimenting with them.
- After making any open-source contribution, we get immediate feedback regarding our developmental work.

- While doing open-source contributions, we interact with like-minded developers from all over the world and build connections along the way.
- As we get more closer to the open-source community, we get to know much more about our field of interest and other related fields.
- The most important aspect of open-source contributions is It may fetch us a job in our field of interest.

Free Software:

"Free software" means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. The term "free software" is sometimes misunderstood—it has nothing to do with price. It is about freedom.

Advantages:

- Cost: Free software is typically free to use, modify and distribute.
- Freedom: Free software is often accompanied by a set of ethical principles that promote users' freedom to use, study, modify, and share the software.
- Collaboration: Free software often encourages collaboration among developers and users, leading to faster development and better quality software.
- Transparency: Free software is often developed in a transparent way, with the source code and development process available for public scrutiny.
- Flexibility: Free software can be used on a wide range of platforms and devices.

Disadvantages:

- Support: While free software does have a community of developers and users, it may not always have the same level of professional support as commercial software.
- Compatibility: Free software may not always be compatible with other software applications and hardware devices.
- Security: Because free software is available for everyone to use and modify, it may be easier for malicious actors to identify and exploit vulnerabilities.
- Complexity: Free software can be more complex and difficult to use than commercial software, especially for non-technical users.
- Documentation: Free software may not always have the same level of documentation and user guides as commercial software.

Open Source Software:

Open Source Software is something that you can modify as per your needs, and share with others without any licensing violation burden. When we say Open Source, the source code of the software is available publicly with Open Source licenses like GNU (GPL) which allows you to edit the source code and distribute it. Read these licenses and you will realize that these licenses are created to help us.

1. Coined by the development environments around software produced by open collaboration of software developers on the internet.
2. Later specified by the Open Source Initiative (OSI).
3. It does not explicitly state ethical values, besides those directly associated with software development.

Advantages:

- Cost: Open source software is typically free to use, modify and distribute.

- Customization: The source code of open source software is available to everyone, allowing users to modify and customize it to suit their needs.
- Community support: Open source software often has a large community of developers and users who contribute to its development and provide support.
- Transparency: The source code of open source software is open for everyone to see, making it easier to identify and fix bugs and vulnerabilities.
- Flexibility: Open source software can be used on a wide range of platforms and devices.

Disadvantages:

- Support: While open source software does have a large community of developers and users, it may not always have the same level of professional support as commercial software.
- Compatibility: Open source software may not always be compatible with other software applications and hardware devices.
- Security: Because the source code of open source software is available to everyone, it may be easier for malicious actors to identify and exploit vulnerabilities.
- Complexity: Open source software can be more complex and difficult to use than commercial software, especially for non-technical users.
- Documentation: Open source software may not always have the same level of documentation and user guides as commercial software.

Similarities:

- Both free software and open source software have access to the source code, allowing users to modify and improve the software.
- Both types of software often rely on a community of users and developers to provide support and contribute to the development of the software.
- Both free software and open source software are often distributed under open licenses, allowing users to use, modify, and distribute the software without restrictions.