- **Programming languages:**
  Programming languages acts as a mediator between user and computer, passing on the information and instructions.

  Programming languages are a communication medium through which user can interact with the computer system using statements. Statements are such type of sentences which can produce one value at a time – True or False. Statements are also known as “Proposition”.

  The rules to write correct statements is called “Syntax”. That means, the grammar of programming languages to write appropriate statement is called Syntax.
- **Machine language:**
  Machine language is the 1st generation language which is written in pure binary form. Machine language is also called Low level language.

  The lowest level component of computer system is hardware, the programming language which has capability to handle lowest level component is called Low level programming language. Machine language has the capacity to directly handle the lowest level component of computer system, so it is called low-level programming language.

  <u>Properties of Machine language:</u>
  - Must be written in pure binary form.
  - Execution speed is maximum because any type of conversion is not needed.
  - It is very difficult to write and debug.
  - Understanding is very difficult.

- It is not portable, which means, a single program cannot be executed in different computer systems.
- Execution speed of machine language program is maximum, so some devices are replaced by machine level program, which is called firmware.

- **Assembly language:**
  Assembly language is also called as 2nd generation programming language. This programming language uses alpha-numeric symbols to write the instruction of a program. Meaningful and easy to remember symbols are used in the instruction, for example- "ADD" for addition, "SUB" for subtraction, etc.

  These symbols are called "Mnemonics". A program written is mnemonics is called assembly language. It also machine depend language.

  Properties of Assembly language:
  - Writing and debugging is easier than machine language.
  - Introduction of alphanumeric codes.
  - Execution speed is comparatively slow due to need of compilation process.
  - It is also non-portable that means, even assembly language programs cannot run on multiple machines without modification.
- **High level language:**
  High level languages were introduces after 2nd generation languages and they are categorized as 3rd, 4th or 5th generation languages. These programming languages uses English related words to write and understand the codes more easily. Due to use of English words, learning new programming languages is also easy.

There are multiple high level languages which needs to compiled or interpreted before execution. Programs written in high level languages are portable and can run different machines without any changes.

Properties of High level language:

- Writing and debugging is easier than ever due to use of English words and different programming paradigms.
- Execution is slow as compared to machine language or assembly language due to the need to be compiled or interpreted.
- Programs written in these programming languages can be run of different machines without any changes as these are portable.

- **Compilers:**
  A Compiler is a software that typically takes a high level language (Like C++ and Java) code as input and converts the input to a lower level language at once. It lists all the errors if the input code does not follow the rules of its language. This process is much faster than interpreter but it becomes difficult to debug all the errors together in a program.

  A compiler is a translating program that translates the instructions of high level language to machine level language. A program which is input to the compiler is called a Source program. This program is now converted to a machine level language by a compiler is known as the Object code.

  Properties of Compiler:

  - It is a piece of software which converts programs written in high level languages to machine language.

- It can convert whole program at once and if there is any error in the syntax, it lists them all at once.
- Debugging is bit tough as it gives all error at once as compared to interpreter.
- It is fast as compared to interpreters.

- **Interpreters:**
  All high level languages need to be converted to machine code so that the computer can understand the program after taking the required inputs. The software by which the conversion of the high level instructions is performed line-by-line to machine level language, other than compiler and assembler, is known as INTERPRETER.

  If an error is found on any line, the execution stops till it is corrected. This process of correcting errors is easier as it gives line-by-line error but the program takes more time to execute successfully. It translates source code into some efficient intermediate representation and immediately execute this. Source programs are compiled ahead of time and stored as machine-independent code, which is then linked at run-time and executed by an interpreter.

  It helps the programmer to find out the errors and to correct them before the control moves to the next statement. Interpreter system performs the actions described by the high-level program. For interpreted programs, the source code is needed to run the program every time. Interpreted programs run slower than compiled programs.

  <u>Properties of Interpreters:</u>
  - It is a piece of software which converts programs written in high level languages to machine language.

- It cannot convert whole program at once. It checks each line one-by-one and then converts it to machine language.
- Debugging is easier than compiled programming languages as user gets and error message for the same line.
- It is slow as compared to compilers.

- **Assemblers:**
  The Assembler is used to translate the program written in Assembly language into machine code. The source program is an input of an assembler that contains assembly language instructions. The output generated by the assembler is the object code or machine code understandable by the computer.

  Assembler is basically the 1st interface that is able to communicate humans with the machine. We need an Assembler to fill the gap between human and machine so that they can communicate with each other. Code written in assembly language is some sort of mnemonics (instructions) like ADD, MUL, MUX, SUB, DIV, MOV and so on. And the assembler is basically able to convert these mnemonics in Binary code.

  <u>Properties of Assemblers:</u>
  - It is a piece of software which converts programs written in assembly languages to machine language.
  - It is an interface between user and machine.
- **Information system:**
  Information systems are a set of interconnected elements working together to collect, process, store, and distribute information to help coordination, visualization in an organization, analysis, and decision-making.

  The Information system can be defined as a collection of software, hardware, and telecommunications network that people develop

and use to gather, create, and distribute useful data, mainly in organizational settings.

In other words, an information system means a collection of interrelated components which work together to gather, process, store, and break down the information to help decision making.

- **Centralized processing:**
  Centralized systems are systems that use client/server architecture where one or more client nodes are directly connected to a central server. This is the most commonly used type of system in many organizations where a client sends a request to a company server and receives the response.

  Characteristics of Centralized System –

  - Presence of a global clock:
    As the entire system consists of a central node (a server/ a master) and many client nodes (a computer/ a slave), all client nodes sync up with the global clock (the clock of the central node).
  - One single central unit:
    One single central unit which serves/coordinates all the other nodes in the system.
  - Dependent failure of components:
    Central node failure causes the entire system to fail. This makes sense because when the server is down, no other entity is there to send/receive responses/requests.
- **Decentralized processing:**
  In decentralized systems, every node makes its own decision. The final behaviour of the system is the aggregate of the decisions of

the individual nodes. Note that there is no single entity that receives and responds to the request.

Characteristics of Decentralized System –

- Lack of a global clock:
  Every node is independent of each other and hence, has different clocks that they run and follow.
- Multiple central units (Computers/Nodes/Servers):
  More than one central unit which can listen for connections from other nodes.
- Dependent failure of components:
  One central node failure causes a part of the system to fail; not the whole system.

- **Distributed process:**
  Distributed processing is a computing process where operations are partitioned across several computers connected via a network. The goal of distributed processing is to provide faster and more reliable service than can be achieved by a single machine.

  Very often, datasets are too big to fit on one machine. Distributed data processing breaks down these large datasets and stores them across multiple machines or servers, improving data management. It rests on Hadoop Distributed File System (HDFS). A distributed data processing system has a high fault tolerance. If one server in the network fails, you can reallocate data processing tasks to other available servers, which is not a very time-consuming job.
- **Management processing modes:**
  The method of data processing used will determine the response time to a query and how reliable the output is. For instance, in a

situation where availability is crucial, such as a stock exchange portal, transaction processing should be the preferred method.

It is important to note the difference between data processing and a data processing system. Data processing refers to the rules by which data converts into useful information. A data processing system is an application optimized for a specific type of data processing. For instance, a timesharing system is designed to run timesharing processing optimally. You can use it to run batch processing, too.

- **Uniprocessor:**
  A uniprocessor system is defined as a computer system that has a single central processing unit that is used to execute computer tasks. As more and more modern software is able to make use of multiprocessing architectures, such as SMP and MPP, the term uniprocessor is therefore used to distinguish the class of computers where all processing tasks share a single CPU. Most desktop computers are shipped with multiprocessing architectures since the 2010s. As such, this kind of system uses a type of architecture that is based on a single computing unit. All operations (additions, multiplications, etc.) are thus done sequentially on the unit.
- **Multiprocessor:**
  A multiprocessor is a computer system with two or more central processing units (CPUs), with each one sharing the common main memory as well as the peripherals. This helps in simultaneous processing of programs.

  The key objective of using a multiprocessor is to boost the system's execution speed, with other objectives being fault tolerance and application matching.

A good illustration of a multiprocessor is a single central tower attached to two computer systems. A multiprocessor is regarded as a means to improve computing speeds, performance and cost-effectiveness, as well as to provide enhanced availability and reliability.

Benefits of using a multiprocessor include:

- Enhanced performance
- Multiple applications
- Multiple users
- Multi-tasking inside an application
- High throughput and/or responsiveness
- Hardware sharing among CPUs

- **Batch processing:**
  As the name suggests, batch processing is when chunks of data, stored over a period of time, are analysed together or in batches. Batch processing is required when business owners and data scientists require a large volume of data to analyse for detailed insights. For example, sales figures will typically undergo batch processing, allowing businesses to use data visualization features like charts, graphs, and reports to derive value from data. Since a large volume of data is involved, the system will take time to process it. Processing the data in batches saves on computational resources.

  You might prefer batch processing over real-time processing when accuracy is more important than speed. Additionally, you can measure the efficiency of batch processing in terms of throughput. Throughput is the amount of data processed per unit of time.

- **Real time processing:**
  Real-time processing is the process of computing data as soon as it is generated or received. It's a form of distributed processing that allows you to capture and analyse incoming data streams in real-time, allowing you to act quickly on the insights given by the analysis.

  Real-time processing is that you use it in situations where you expect output in real-time. Real-time processing computes incoming data as quickly as possible. If it encounters an error in incoming data, it ignores the error and moves to the next chunk of data input coming in. GPS-tracking applications are the most common example of real-time data processing.
- **Time sharing processing:**
  Time-sharing, in data processing, method of operation in which multiple users with different programs interact nearly simultaneously with the central processing unit (CPU) of a large-scale digital computer. Because the CPU operates substantially faster than most peripheral equipment (e.g., video display terminals and printers), it has sufficient time to solve several discrete problems during the input/output process. Even though the CPU addresses the problem of each user in sequence, access to and retrieval from the time-sharing system seems instantaneous from the standpoint of remote terminals since the solutions are available to them the moment the problem is completely entered.

  Commonly used time-sharing techniques include multiprocessing, parallel operation, and multiprogramming. Also, many computer networks organized for the purpose of exchanging data and resources are centred on time-sharing systems.

- **Electronic mail:**
  Email (electronic mail) is the exchange of computer-stored messages from one user to one or more recipients via the internet. Emails are a fast, inexpensive and accessible way to communicate for business or personal use. Users can send emails from anywhere as long as they have an internet connection, which is typically provided by an internet service provider.

  Email is exchanged across computer networks, primarily the internet, but it can also be exchanged between both public and private networks, such as a local area network. Email can be distributed to lists of people as well as to individuals.
- **Tele-text conferencing:**
  Teletext or broadcast teletext is a standard for displaying text and rudimentary graphics on suitably equipped television sets. Teletext is a system that links a computer with a television by which text and graphic information can be transmitted on a one way basis to home viewers. Indian television join the advance nations in 1985, it started the teletext known as index service to telecast the latest news and information on the stock exchange air lines, railway timings, weather information etc. In the system a fix number of pages are broadcast sequentially but continuously, each page consist of television screen, display of text and graphic shapes.