BBA

Computer & IT Applications - I

UNIT I:

A computer is a fast electronic device that processes raw data, performs arithmetic and logical operations, and produces output. Nowadays, the computer has become a part of our life and can be used in most fields. In a computer, there are various characteristics of computer systems depending on their size, capacity, and specifications. But, the major characteristics of the computer can be classified into Speed, Accuracy, Diligence, Versatility, Reliability, Consistency, Memory, Storage Capacity, Remembrance Power, and Automation's.

What is Computer?

A computer system is a basic and functional computer that includes all the hardware and software that are required to make it functional for the user. It is an electronic device that accepts the data as input, processes the data in a pre-determined way and then communicates the result as output on the screen. CPU which stands for Central Processing Unit is the brain of the computer. It is an electronic circuit within a computer that executes instructions that make up a computer program. Various blocks of CPU are-

- It carries out the operations of arithmetic and logical expressions.
- The timing and Control Unit controls the entire operation being performed on the computer system.
- The register is a very small amount of very fast memory that is built into the CPU in order to store the current data and instructions that are being performed by the CPU.
- Unit is the storage unit of the computer system which is used to store the program statement and data.
- Input and Output unit sends and receives data to display it to the user.

Characteristics of Computer System

1. Speed

Executing mathematical calculation, a computer works faster and more accurately than human. Computers have the ability to process so many millions (1,000,000) of instructions per second. Computer operations are performed in micro and nano seconds. A computer is a time saving device. It performs several calculations and tasks in few seconds that we take hours to solve. The speed of a computer is measure in terms of GigaHertz and MegaHertz.

2. Diligence

A human cannot work for several hours without resting, yet a computer never tires. A computer can conduct millions of calculations per second with complete precision without stopping. A computer can consistently and accurately do millions of jobs or calculations. There is no weariness or lack of concentration. Its memory ability also places it ahead of humans.

3. Reliability

A computer is reliable. The output results never differ unless the input varies. The output is totally depend on the input. When an input is the same the output will also be the same. A computer produces consistent results for similar sets of data, if we provide the same set of input at any time we will get the same result.

4. Automation

The world is quickly moving toward AI (Artificial Intelligence)-based technology. A computer may conduct tasks automatically after instructions are programmed. By executing jobs automatically, this computer feature replaces thousands of workers. Automation in computing is often achieved by the use of a program, a script, or batch processing.

5. Versatility

Versatility refers to a capacity of computer. Computer perform different types of tasks with the same accuracy and efficiency. A computer can perform multiple tasks at the same time this is known as versatility. For example, while listening to music, we may develop our project using PowerPoint and WordPad, or we can design a website.

6. Memory

A computer can store millions of records. These records may be accessed with complete precision. Computer memory storage capacity is measured in Bytes, Kilobytes (KB), Megabytes (MB), Gigabytes (GB), and Terabytes (TB). A computer has built-in memory known as primary memory.

7. Accuracy

When a computer performs a computation or operation, the chances of errors occurring are low. Errors in a computer are caused by human's submitting incorrect data. A computer can do a variety of operations and calculations fast and accurately.

Application /Uses of Computer in Business

- **Inventory Management:** Computers helps in tracking inventory levels and automating reordering processes. It also helps in monitoring stock movement. This assist in overstocking while optimizing supply chain efficiency.
- **E-commerce:** Computers helps in online selling and purchasing through e-commerce platforms. Businesses can set up online stores, process orders, and handle payment transactions can be done by computer.
- **Business Analytics:** Computers are used to analyse data and perform tasks from various sources. Computer can help in business intelligence tools. Making decisions by visualizing and interpreting data trends can be done by computer..
- **Human Resources Management:** Computers can help in storing employee datas, recruitment, calculate performance, training, and payroll processing.
- **Data Management and Storage:** Through computer databases and file management systems help organize and get all information correctly. Computers are used manage large numbers of business data, save customer information, financial records, inventory data, and more.

- **Communication:** Computers help business to communicate the outside organization. Emails, video conferencing, and collaboration sites enable communication among employees, team members, and clients.

- **Accounting and Financial Management:** Computers are necessary for accounting tasks, like financial analysis, budgeting, and payroll management. Accounting software helps in financial operations and store accurate record safely.

- **Sales and Marketing:** Computers assist in customer relationship management (CRM) systems, tracking sales record, and marketing. Computers help businesses manage leads and handle customer interactions.

- **Data Analysis and Reporting:** Computers analyze business data to create reports and guide strategic decisions. Data visualization tools through computer make it easier to understand information.

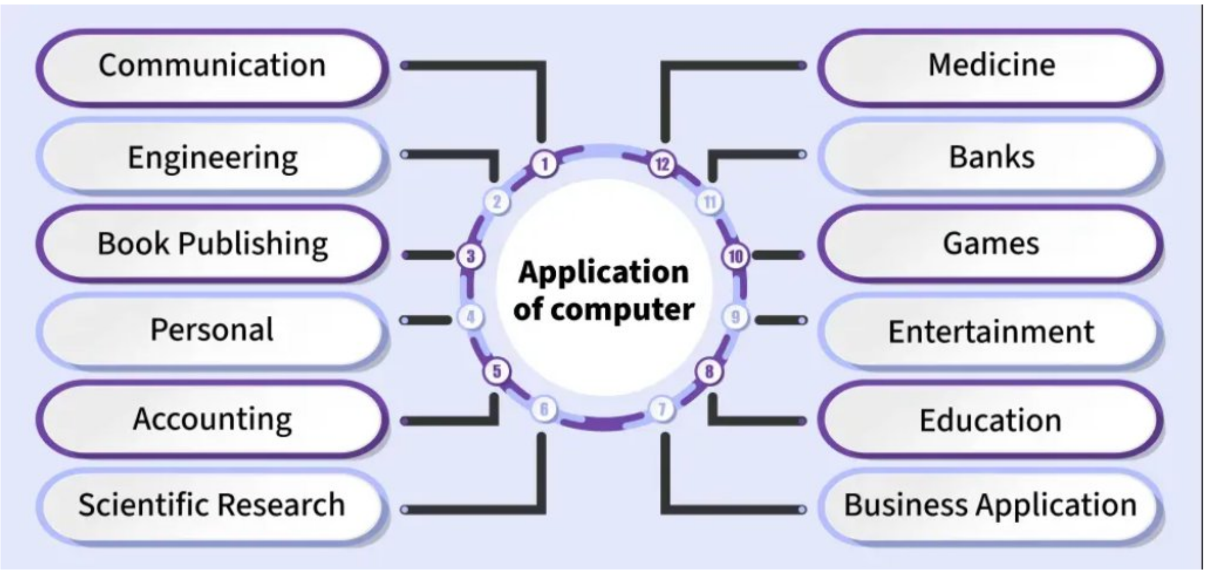
- **Remote Work and Telecommuting:** Computers helps in remote work by providing employees access to company applications, and data from any remote locations.

- **Security and Cyber security:** Computers helps in many security measures like firewalls, encryption, and more to secure business data from cyber-attacks.

Basic Applications of Computers

Computers are used in every field of life, such as homes, businesses, educational institutions, research organizations, the medical field, government offices, entertainment, etc.Today we cannot imagine growing our technology without computers.

The various field where the computer is very essential are:



1. **Science**

Today computer is the best-suited machine for collecting, analyzing, classifying, and storing data. It becomes the most essential medium to spread knowledge internally and internationally. It allows scientists from different locations to work together and share ideas on the same project.

2. **Defence System**

Computers are used to track airplanes, missiles, tanks, and different kinds of weapons. Once the radar system tracks a missile and artificial intelligence is programmed to target a missile and destroy it before it comes on the surface. It also used for GPS tracking, controlling defence vehicles, records of all members of the military.

3. **Medical**

Computers are used to record patients' information, monitoring heart rate, oxygen level, and blood pressure. To conduct various surgeries junior doctors get the help of another professional doctor by web conferencing.

4. **Education**

Computers are very crucial for online classes, download study material on the internet and interact with fellow learners. Computers are also used to track student attendance and learning strategies.

5. **Banking**

A computer help in storing several account holder details on a bank server. All transactions such as deposits and withdrawals perform by a computer. A banking company can easily monitor all ATMs and passbook printing machines.

6. **Government Sectors**

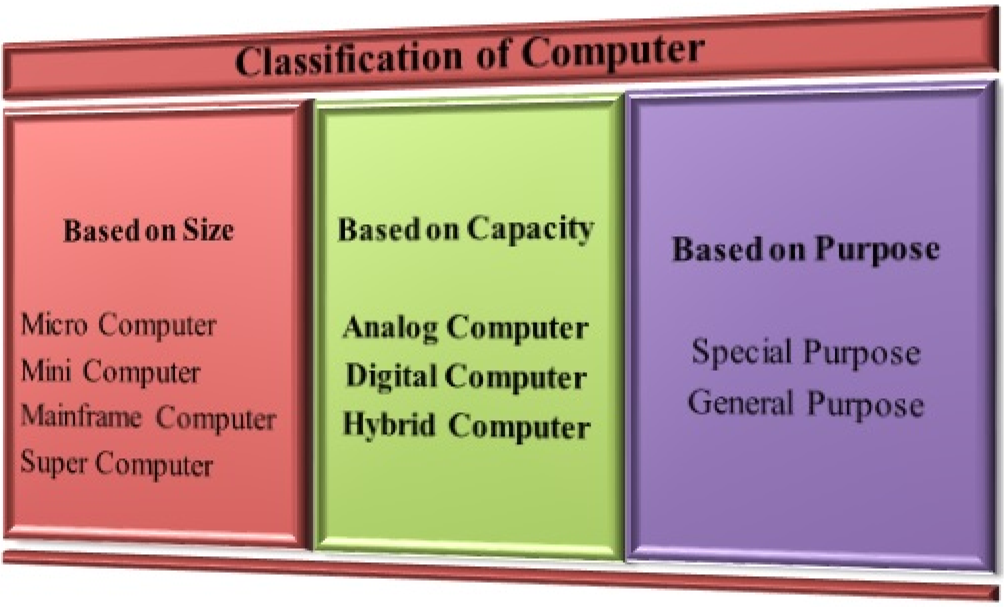
Government can easily monitor government sectors such as road services, railway, development, and other rising funds. The information of every citizen is stored on the server through the computer.

7. **Entertainment**

We can play various interesting video games using a computer. We can watch movies, TV shows, and reality shows on the computer. A computer is also used to create sarcastic memes and make us happy.

A computer can be classified based on its size, capacity, and purpose. The following diagram illustrates different types of computers as per their size, capacity, and purpose.

Classification of Computer



Computer's classification based on Size

As per the size, a computer can be broadly classified as follows −

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

1. Micro Computer

Microcomputers, also known as personal computers (PCs), are a type of computer designed for individual use. They are distinguished by their compact dimensions, small size, processing power, compatibility, internet connectivity, portability, low price, and versatility. In the 1970s and 1980s, microcomputers gained popularity and became more popular in the modern computing era.

Fig: Micro Computer

- **Size** − Microcomputers are small in size. These are portable.
- **Example** − Some of the popular microcomputers are laptops and desktops, standard PCs, mobile phones, and notebooks.

- **Why microcomputer ?**

Microcomputers have become an important part of modern life. They have had a big impact on society, companies, education, and related areas.

- **Uses of Microcomputers** − Microcomputers are most widely used in education and learning, entertainment and media, innovation and creativity, research and science, healthcare and medicine, home automation, remote work, and e-commerce and online shopping.

2. Minicomputer

A minicomputer is a type of computer that is smaller in size than large computers. It possesses all the capabilities of a large computer. Hence, it is a midsize multi-processing system capable of supporting up to 250 users simultaneously.

Fig: Mini Computer

- **Size** − Its size falls between mainframes and microcomputers. It is larger than mainframe computers and smaller than microcomputers.
- **Example** − Some of the popular minicomputers are the PDP-11, IBM's AS/400e, Honeywell 200, and TI-990.
- **Why a Mini Computer?** Mini computers are also known as mini PCs or small-form-factor (SFF) computers. These have impressive computing capabilities, high performance, connectivity options, portability, and versatility features.
- **Uses of Minicomputers** − Minicomputers are most widely used in scientific computations, engineering, business transaction processing, file handling, and database management.

3. Mainframe computer

The mainframe is very large and is an expensive computer capable of supporting hundreds or even thousands of users simultaneously. The mainframe executes many programmes concurrently and supports simultaneous execution of programmes.

Fig: Mainframe Computer

- **Size** − Mainframe computers can vary in size; their size generally depends on their specifications and the specific model being considered.
- **Example** − Some of the popular mainframe computers are IBM zSeries mainframes (BM z14 and IBM z15), Unisys ClearPath, Fujitsu GS21 Series, and Hitachi VOS3.
- **Why a mainframe computer?**

The processing capacity of mainframes is frequently measured in MIPS (million instructions per second) or other units. This enables them to process a large volume of transactions and perform extensive data processing.

- **Uses of the Mainframe** − Mainframe computers are most widely used in finance, government, healthcare, and more.

4. Supercomputer

A supercomputer is a special type of computer that is more powerful and capable of high-performance computing. It is specifically designed to compute complex and intensive tasks that regular computers cannot do efficiently.

- **Size** − Supercomputers can vary in size, from small clusters of computers to massive installations. A supercomputer may contain 10, 100, 1000, or more computers that all work together.

Fig: Super Computer

- **Example** − Some of the popular supercomputers are Fugaku, Google Sycamore, Baidu's quantum supercomputer, and Sierra.

- **Why Supercomputer?**
  - A supercomputer's processing speed is exceptional and can perform billions of calculations per second. Multiple processors work in parallel mode to execute tasks, which makes processing powerful.
  - Supercomputers are specially built using specialised hardware like GPUs (Graphics Processing Units) or TPUs (Tensor Processing Units), which are used in graphics rendering or machine learning tasks.
  - Supercomputers represent the pinnacle of computing power, and these are very expensive and are employed for specialised applications.
- **Uses of the Supercomputer** – Supercomputers are most widely used in scientific research, data analysis, weather forecasting, scientific simulations, graphics, fluid dynamic calculations, nuclear energy research, electronic design, and the analysis of geological data.

Computer's classification based on Capacity

As per the capacity, a computer can be broadly classified as follows –

1. Analog Computer
2. Digital computer
3. Hybrid computer

1. Analog Computer

A computer that uses physical means like mechanical or hydraulic components to do the computation rather than electronic circuits is called an analogue computer. These computers work with continuous data and can manage physical quantities efficiently. They are particularly good at solving differential equations and simulating dynamic systems.

Fig: Analog computer

In lieu of numbers, an analogue computer performs arithmetic operations based on measurable quantities, such as mechanical movement or the rotation of gears. In analogue computers, data is processed as continuous signals for its operation, whereas in digital computers, data is transmitted as discrete signals (or discontinuous signals).

2. Digital Computer

A digital computer is a type of computer that represents and processes data using discrete, distinct values.

Fig: Digital computer

In digital computers, data is processed using binary numbers 0 and 1. These computers are designed to perform arithmetic calculations and complex data processing and manipulation. The main components of a digital computer are input, processing, and output.

3. Hybrid Computer

A hybrid computer is a type of computer system that integrates the features and capabilities of both analogue and digital computers. This integration allows the hybrid computer to perform various tasks efficiently by leveraging the strengths of both digital and analogue technologies.

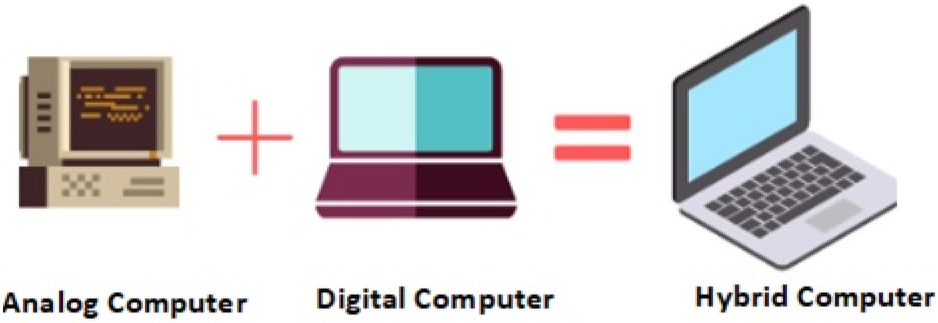


Fig: Hybrid computer

The main components of a hybrid computer are the analogue and digital components −

- **Analog Component** − Analogue components in a hybrid computer can process real-world data like voltage, current, temperature, pressure, etc. using analogue circuits and components.
- **Digital Component** − Digital computers work with discrete data and are based on binary numbers (0s and 1s). Digital components in a hybrid computer provide the computational power to perform complex calculations and control the overall operation of the system.

Computer's classification based on Purpose

As per the capacity, a computer can be broadly classified as follows −

1. Special Purpose
2. General Purpose

1. Special Purpose Computer

A computer that is designed and optimised for a specific task or set of tasks is called a special purpose computer (SPC). SPCs are designed to excel at a single or limited set of functions, frequently with a high degree of efficiency, speed, and accuracy.

Fig: Special Purpose Computer

Some of the following popular SPCs are:

- **Embedded Systems** − These systems are integrated with devices to control specific functions. For example, a car's engine control unit and microwave ovens
- **Digital Signal Processors** − These are commonly used in applications like audio processing, image compression, and telecommunications.
- **Automated Teller Machines** − ATMs are special-purpose computers designed specifically for banking transactions and interactions with customers.
- **Medical Equipment** − Machines like MRI and CT scanners are specialised computers used for capturing and processing medical images.
- **Spacecraft Computers** − Computers used in spacecraft have to operate in extreme conditions and are optimised for the demands of space missions.

2. **General Purpose Computer**

A computer that is designed to perform a wide range of tasks and functions is called A General Purpose Computer (GPC). A GPC is versatile and can be used for various purposes by running different software and applications.

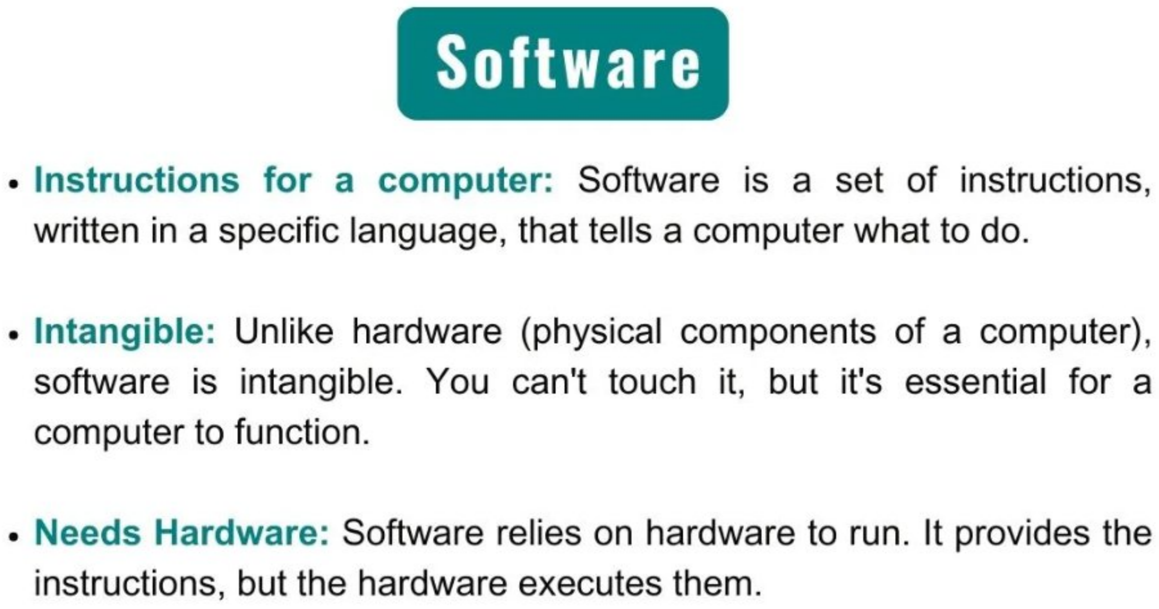
Fig: General Purpose Computer

Some of the following popular GPCs are as −

- **Turing Completeness** − A GPC can simulate any algorithm or computation that can be explored algorithmically.
- **Programmability** − GPCs can run different applications.

- **General-Purpose Operating System** – GPCs like Windows, macOS, or Linux that provide an interactive user interface and manage hardware resources, enabling the execution of various application programmes.
- **Input and Output Capabilities** – GPCs have input and output devices (keyboard, mouse, monitor, etc.) that permit users to interact with the system and receive feedback.

Software is a set of instructions that tell a computer what to do. These instructions are source codes written in programming languages such as Python, Java, or C++. The source code is then converted into machine-readable object code or executable files, which can be run on a **computer** or other electronic devices.

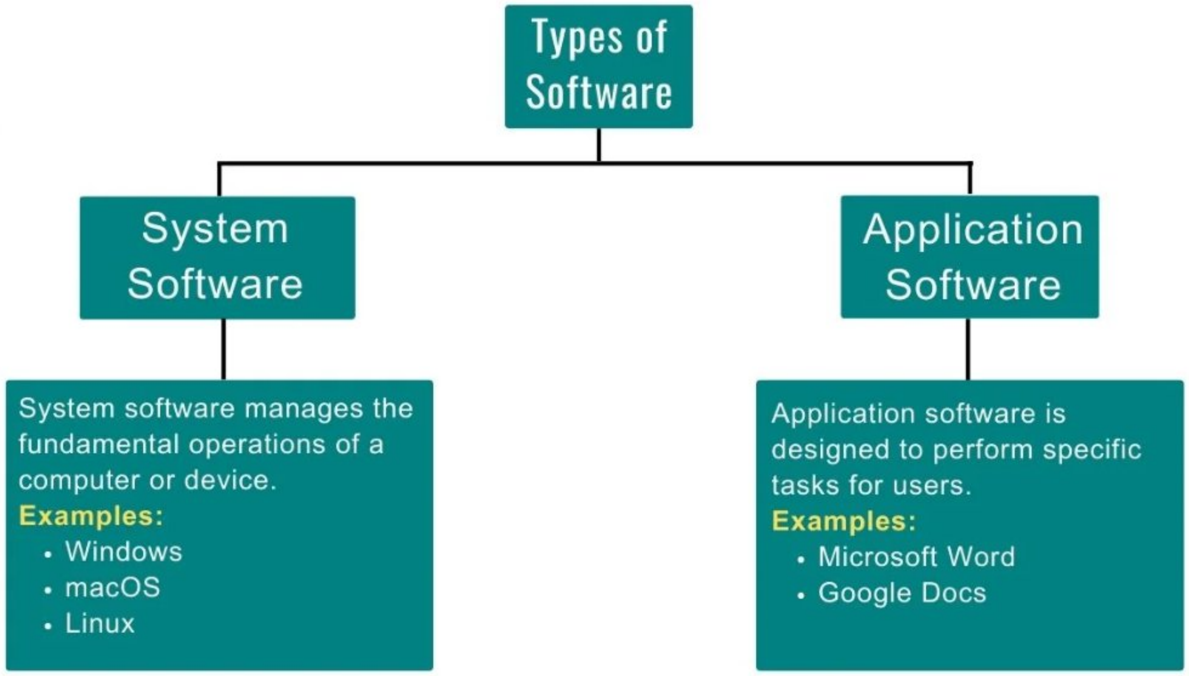


Software

- **Instructions for a computer:** Software is a set of instructions, written in a specific language, that tells a computer what to do.
- **Intangible:** Unlike hardware (physical components of a computer), software is intangible. You can't touch it, but it's essential for a computer to function.
- **Needs Hardware:** Software relies on hardware to run. It provides the instructions, but the hardware executes them.

Types of Software

Software can be broadly categorized into three main types:


Types of Software
System Software
Application Software
System software manages the fundamental operations of a computer or device.
Examples:
• Windows
• macOS
• Linux
Application software is designed to perform specific tasks for users.
Examples:
• Microsoft Word
• Google Docs

- **System Software**: This type of software manages the fundamental operations of a computer or device. Examples include operating systems (e.g., Windows, macOS, Linux) and device drivers that control hardware components.

- **Application Software:** Also known as productivity software, application software is designed to perform specific tasks for users. Examples include word processors (e.g., Microsoft Word, Google Docs), spreadsheet programs (e.g., Excel, Sheets), media players, and web browsers.

- **Embedded Software:** This type of software is integrated into non-computing devices, such as consumer electronics, automobiles, and industrial machinery. Embedded software controls the functionality and behavior of these devices, enabling features like touchscreens, automatic transmissions, and safety systems.

Software Components

Software is typically composed of three main components:

1. **Source Code:** This is the human-readable instructions written in programming languages by developers.

2. **Object Code:** Once the source code is compiled or interpreted, it becomes machine-readable object code.

3. **Executable Files:** These files contain the object code and can be executed or run on a computer.

Software Applications

Software applications can be broadly classified into three categories based on their intended use:

1. **Productivity Software:** These applications are designed to assist users in completing tasks such as word processing, spreadsheet management, and presentation creation. Examples include Microsoft Office Suite and Google Workspace.

2. **Entertainment Software:** This category includes various forms of multimedia software, such as video games, media players, and streaming applications.
3. **Educational Software:** These applications are designed for learning and teaching purposes, ranging from educational games to online courses and reference materials.

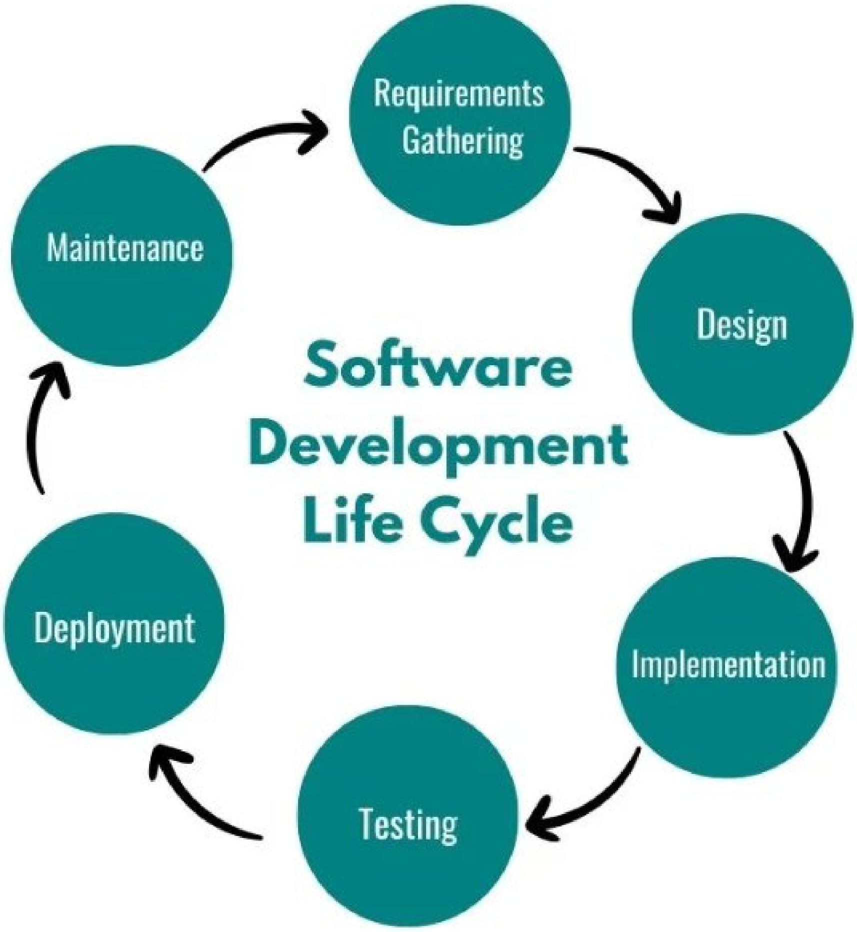
Software Characteristics

Effective software should possess several key characteristics including:

- **User Interface:** How users interact with and control the software, such as a graphical user interface (GUI) or a command-line interface (CLI).
- **Functionality:** The set of features and capabilities provided by the software to meet user requirements.
- **Reliability:** The ability of the software to perform its intended functions consistently and accurately.
- **Performance:** The measure of how efficiently the software utilizes system resources, such as memory, processing power, and storage.

Software Development Life Cycle

From Idea to Implementation Creating software is a complex process that involves several stages, collectively known as the Software Development Life Cycle (SDLC). The SDLC typically includes the following steps:



1. **Requirements Gathering:** Understanding the needs and objectives of the software project.
2. **Design:** Planning the software's architecture, user interface, and overall structure.
3. **Implementation:** Writing the actual code that brings the software to life.
4. **Testing:** Ensuring the software functions as intended and identifying and fixing any defects or errors.
5. **Deployment:** Releasing the software for use by end-users.
6. **Maintenance:** Updating and improving the software over time to address new requirements or fix issues.

OR

What is Software?

Software is a set of instructions, programs, and data that tell a computer how to perform specific tasks. It acts as an interface between the user and the hardware. Without software, computer hardware cannot function properly.

Types of Software

Software can be broadly divided into the following categories:

1) System Software

- Helps run and manage the computer hardware and provides a platform for running application software.
- **Examples:**
  - Operating Systems (Windows, Linux, macOS)
  - Utility Programs (Antivirus, Disk Cleanup tools)
  - Device Drivers (Printer driver, Display driver)

2) Application Software

- Designed for end-users to perform specific tasks or solve particular problems.
- **Examples:**
  - Word Processors (MS Word)

  - Web Browsers (Google Chrome, Firefox)
  - Media Players (VLC, Windows Media Player)

3) Programming Software (Development Tools)

- Used by programmers to develop, test, and maintain other software.
- **Examples:**
  - Compilers (GCC, Turbo C)
  - Interpreters (Python Interpreter)
  - IDEs (Visual Studio, Eclipse)

4) Middleware (Optional Category)

- Acts as a bridge between different software or between application and system software.
- **Example:** Database middleware connecting applications to databases.

5) Firmware (Optional Category)

- Software embedded into hardware devices to control them.
- **Examples:** BIOS in computers, software inside smart TVs or routers.

Summary:

- **System Software:** Runs the computer.
- **Application Software:** Helps users perform tasks.
- **Programming Software:** Helps develop software.
- **Middleware & Firmware:** Provide connectivity and control.

Compiler and Interpreter

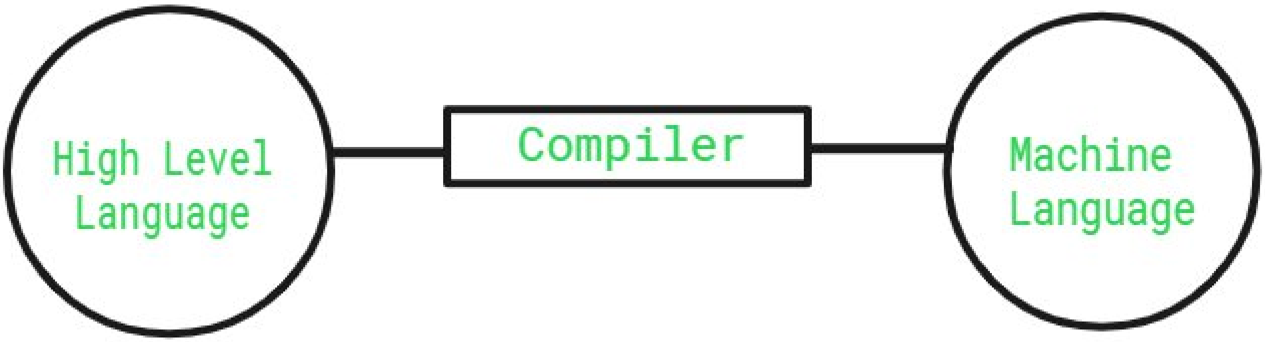
The Compiler and Interpreter, both have similar works to perform. Interpreters and Compilers convert the Source Code (HLL) to Machine Code (understandable by Computer). In general, computer programs exist in High-Level Language that a human being can easily understand. But computers cannot understand the same high-level language, so for computers, we have to convert them into machine language and make them readable.

What is a Compiler?

The Compiler is a translator that takes input i.e., High-Level Language, and produces an output of low-level language i.e. machine or assembly language. The work of a Compiler is to transform the codes written in the programming language into machine code (format of 0s and 1s) so that computers can understand.

Role of a Compiler

For Converting the code written in a high-level language (into machine-level language so that computers can easily understand, we use a compiler. Converts basically convert high-level language to intermediate assembly language by a compiler and then assembled into machine code by an assembler.



Compiler

Advantages of Compiler

- Compiled code runs faster in comparison to Interpreted code.
- Compilers help improve the security of Applications.
- Compilers give Debugging tools, which help in fixing errors easily.

Disadvantages of Compiler

- The compiler can catch only syntax errors and some semantic errors .
- Compilation can take more time in the case of bulky code.

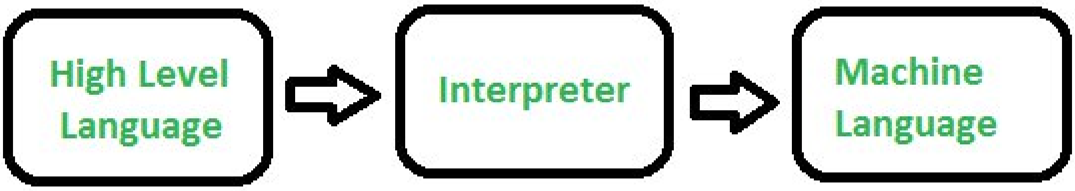
What is an Interpreter?

An Interpreter is a program that translates a programming language into a comprehensible language. The interpreter converts high-level language to an intermediate language. It contains pre-compiled code, source code, etc.

- It translates only one statement of the program at a time.
- Interpreters, more often than not are smaller than compilers.

Role of an Interpreter

The simple role of an interpreter is to translate the material into a target language. An Interpreter works line by line on a code. It also converts high-level language to machine language.



Interpreter

Advantages of Interpreter

- Programs written in an Interpreted language are easier to debug.
- Interpreters allow the management of memory automatically, which reduces memory error risks.
- Interpreted Language is more flexible than a Compiled language.

Disadvantages of Interpreter

- The interpreter can run only the corresponding Interpreted program.
- Interpreted code runs slower in comparison to Compiled code.

Difference between Compiler and Interpreter

Feature	Compiler	Interpreter
Translation	Translates entire code at once	Translates code line by line
Speed	Faster execution	Slower execution
Error Reporting	Shows all errors after compilation	Shows error immediately and stops
Output	Generates an executable file	Does not generate an executable file

Examples of High-Level Languages:

- C, C++
- Java
- Python
- JavaScript
- PHP
- Ruby

Categories of Computer Languages

There are two types of computer languages:

Low-Level Languages: These languages give instructions to a computer in a way that is easily understood by the hardware of the computer. These languages are easier for a computer to understand but difficult for a human understanding. This language is machine-dependent or specific to a given computer. Two low-level languages are Machine Language and Assembly Language.

High-Level Languages: These languages are written in English-like language. Thus, these are easier for a human to understand but difficult for a computer to understand. They can be executed on a machine using a translator. This language is machine-independent. There are many high-level languages eg, C. C++. Java, COBOL, PHP, etc.

Generation of Language

Evolution of Computer Languages

A computer can understand instructions in terms of electric signals as it works on electricity. Electric signals are either ON or OFF, or we can say 1 for ON or 0 for OFF. When computers we invented, firstly instructions were given to it in terms of 1's and 0's. This form of language is known as Machine Language or the first-generation language.

1. **First Generation Language:**

Firstly, computers were given instructions in the forms of 1's and 0's. This language is called Machine Language or first-generation language. A computer was able to understand it directly without any conversion. This language is also known as Machine Language or Binary language. Binary language because only two symbols 1 and 0.

2. **Second Generation Language:**

Machine Language was just 1's and 0's, in which error finding and correcting were very difficult. So we have to develop a second-generation language. Also called an Assembly Language.

In this language, instructions were replaced with some coded terms called mnemonic. So that it becomes a little easier to read, understand, and correct. A computer can understand and work on machine code only. So assembly language needed a special software called Assembler that converts mnemonic into machine language.

3. **Third Generation Languages:**

First, two generations of languages were comparatively easier for a computer to understand, but it was difficult for humans to read, understand and code in them. Thus, came English like programming languages to give instructions to computers. These languages are known as High-Level Languages as they are easier for humans to understand. C, C++, Java, COBOL Pascal, etc. are some High-Level Languages.

4. **Fourth Generation Language:**

Third-generation languages require detailed procedures, but fourth-generation languages just require 'what' do we want from the code rather than 'how to do'. i.e., the procedure. These languages are similar to statements in the human language mainly used in database programming. E.g. Python, Ruby, SQL, MatLab.

SQL-Structured Query Language is one such language, e.g. you just write SELECT ALL department Name FROM EmployeeTable to get all the department names: No detailed program is required.

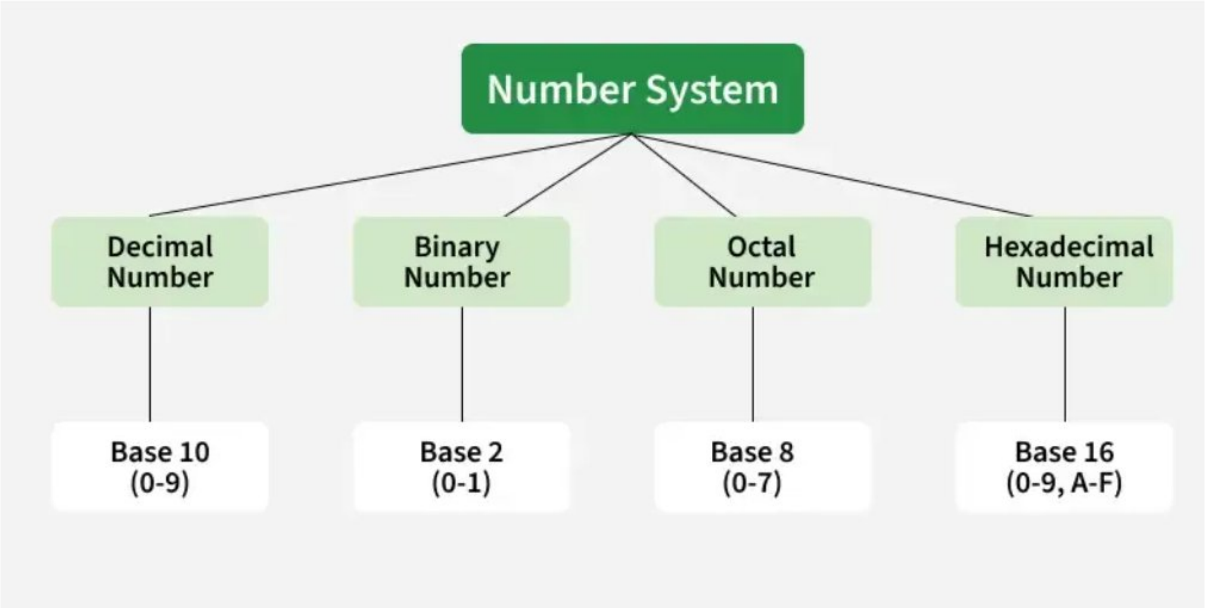
5. **Fifth Generation Languages:**

The fifth-generation languages are more focused on artificial intelligence implementation. These have visual tools to develop a program. Prolog, OPSS Mercury, etc. are some SGLS. E.g. Prolog, OPS5, Mercury etc.

What is a Number?

What is Number System?

- A number system is a way to represent and express numbers using a consistent set of symbols or digits.
- A number system uses a base (or radix) to represent values.
- The base refers to the number of unique digits, including zero that a system uses to represent numbers.
- Most commonly used number systems are Decimal (base-10), Binary (base-2), Octal (base-8), and Hexadecimal (base-16).
- Each system has its own set of rules for representing.
- Digital systems primarily rely on the Binary system for data processing.



Types of Number System

There are four common types of number systems based on the radix or base of the number:

1. Decimal Number System

- The Decimal system is a base-10 number system.
- It uses ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.
- Each digit's place value is a power of 10 (e.g., 100, 101, 102).
- It is the standard system for everyday counting and calculations.

2. Binary Number System

- The Binary system is a base-2 number system.
- It uses two digits: 0 and 1.
- Each digit's place value is a power of 2 (e.g., 20, 21, 22).
- The Binary system is the foundation for data representation in computers and digital electronics.

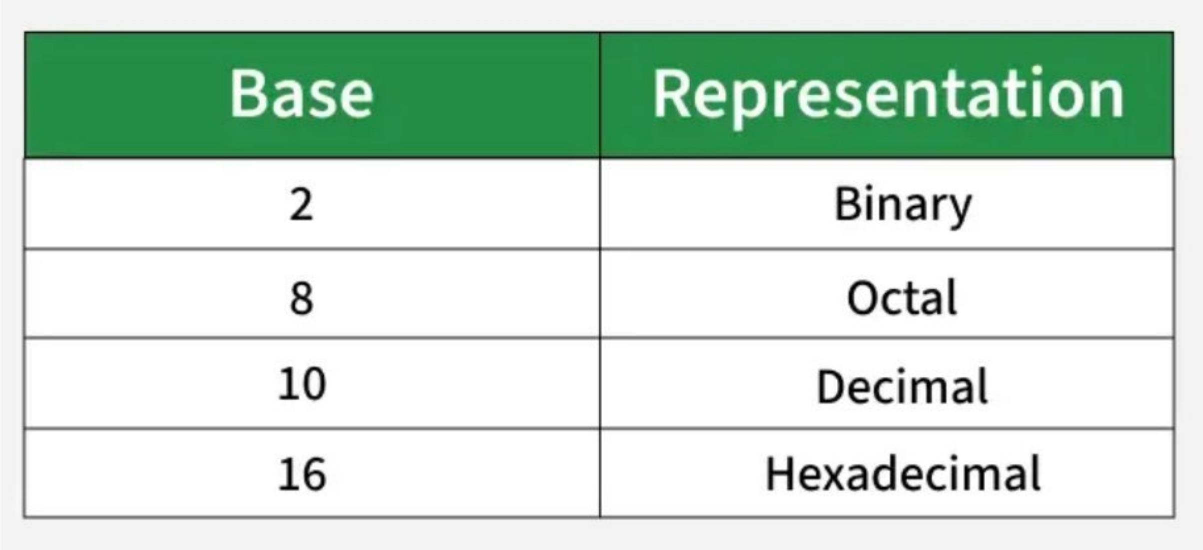
3. Octal Number System

- The Octal system is a base-8 number system.
- It uses eight digits: 0, 1, 2, 3, 4, 5, 6 and 7.
- Each digit's place value is a power of 8 (e.g., 80, 81, 82).
- It is often used to simplify the representation of binary numbers by grouping them into sets of three bits.

4. Hexadecimal Number System

- The Hexadecimal system is a base-16 number system.

- It uses sixteen digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F (where A = 10, B = 11, etc.).
- Each digit's place value is a power of 16 (e.g., 160, 161, 162).
- Hexadecimal simplifies binary by representing every 4 bits as one digit (0-F).



Base	Representation
2	Binary
8	Octal
10	Decimal
16	Hexadecimal

1. Decimal to Binary Number System Conversion

For Integer Part:

- Divide the decimal number by 2.
- Record the remainder (0 or 1).
- Continue dividing the quotient by 2 until the quotient is 0.
- The binary equivalent is the remainders read from bottom to top.

For Fractional Part:

- Multiply the fractional part by 2.
- Record the integer part (0 or 1).
- Take the fractional part of the result and repeat the multiplication.
- Continue until the fractional part becomes 0 or reaches the desired precision.
- The binary equivalent is the integer parts recorded in sequence.

Example: $(10.25)_{10}$

$(868)_{10}$

Decimal to Binary Conversion

For Integer Part (10):

- Divide 10 by 2 → Quotient = 5, Remainder = 0
- Divide 5 by 2 → Quotient = 2, Remainder = 1
- Divide 2 by 2 → Quotient = 1, Remainder = 0
- Divide 1 by 2 → Quotient = 0, Remainder = 1

Reading the remainders from bottom to top gives 1010.

For Fractional Part (0.25):

- Multiply 0.25 by 2 → Result = 0.5, Integer part = 0
- Multiply 0.5 by 2 → Result = 1.0, Integer part = 1

The fractional part ends here as the result is now 0. Reading from top to bottom gives 01.

Thus, the binary equivalent of $(10.25)_{10}$ is $(1010.01)_{2}$.

2. Binary to Decimal Number System Conversion

For Integer Part:

- Write down the binary number.
- Multiply each digit by 2 raised to the power of its position, starting from 0 (rightmost digit).
- Add up the results of these multiplications.
- The sum is the decimal equivalent of the binary integer.

For Fractional Part:

- Write down the binary fraction.
- Multiply each digit by 2 raised to the negative power of its position, starting from -1 (first digit after the decimal point).
- Add up the results of these multiplications.
- The sum is the decimal equivalent of the binary fraction.

Example: $(1010.01)_2$

$1x2_3 + 0x2_2 + 1x2_1 + 0x2_0 + 0x2_{-1} + 1x2_{-2} = 8+0+2+0+0+0.25 = 10.25$

Thus, $(1010.01)_2 = (10.25)_{10}$

3. Decimal to Octal Number System Conversion

For Integer Part:

- Divide the decimal number by 8.
- Record the remainder (0 to 7).
- Continue dividing the quotient by 8 until the quotient is 0.
- The octal equivalent is the remainders read from bottom to top.

For Fractional Part:

- Multiply the fractional part by 8.
- Record the integer part (0 to 7).
- Take the fractional part of the result and repeat the multiplication.
- Continue until the fractional part becomes 0 or reaches the desired precision.
- The octal equivalent is the integer parts recorded in sequence.

Example: $(10.25)_{10}$

For Integer Part (10):

- Divide 10 by 8 → Quotient = 1, Remainder = 2
- Divide 1 by 8 → Quotient = 0, Remainder = 1

Octal equivalent = 12 (write the remainder, read from bottom to top). So, the octal equivalent of the integer part 10 is 12.

For Fractional Part (0.25):

- Multiply 0.25 by 8 → Result = 2.0, Integer part = 2

The fractional part ends here as the result is now 0. So, the octal equivalent of the fractional part 0.25 is 0.2.

The octal equivalent of $(10.25)_{10} = (12.2)_8$

4. Octal to Decimal Number System Conversion

For Integer Part:

- Write down the octal number.
- Multiply each digit by 8 raised to the power of its position, starting from 0 (rightmost digit).
- Add up the results of these multiplications.
- The sum is the decimal equivalent of the octal integer.

For Fractional Part:

- Write down the octal fraction.
- Multiply each digit by 8 raised to the negative power of its position, starting from -1 (first digit after the decimal point).
- Add up the results of these multiplications.
- The sum is the decimal equivalent of the octal fraction.

Example: $(12.2)_8$

$1 \times 8_1 + 2 \times 8_0 + 2 \times 8_{-1} = 8+2+0.25 = 10.25$

Thus, $(12.2)_8 = (10.25)_{10}$

5. Decimal to Hexadecimal Conversion

For Integer Part:

- Divide the decimal number by 16.
- Record the remainder (0-9 or A-F).
- Continue dividing the quotient by 16 until the quotient is 0.
- The hexadecimal equivalent is the remainders read from bottom to top.

For Fractional Part:

- Multiply the fractional part by 16.
- Record the integer part (0-9 or A-F).
- Take the fractional part of the result and repeat the multiplication.
- Continue until the fractional part becomes 0 or reaches the desired precision.
- The hexadecimal equivalent is the integer parts recorded in sequence.

Example: $(10.25)_{10}$

Integer part:

- 10 ÷ 16 = 0, Remainder = A (10 in decimal is A in hexadecimal)

Hexadecimal equivalent = A

Fractional part:

- 0.25 × 16 = 4, Integer part = 4

Hexadecimal equivalent = 0.4

Thus, $(10.25)_{10} = (A.4)_{16}$

6. Hexadecimal to Decimal Conversion

For Integer Part:

- Write down the hexadecimal number.
- Multiply each digit by 16 raised to the power of its position, starting from 0 (rightmost digit).
- Add up the results of these multiplications.
- The sum is the decimal equivalent of the hexadecimal integer.

For Fractional Part:

- Write down the hexadecimal fraction.
- Multiply each digit by 16 raised to the negative power of its position, starting from -1 (first digit after the decimal point).
- Add up the results of these multiplications.
- The sum is the decimal equivalent of the hexadecimal fraction.

Example: $(A.4)_{16}$

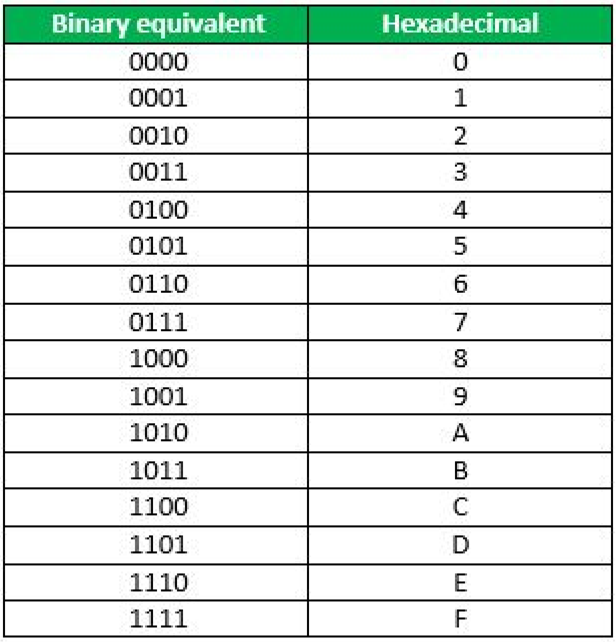
$(A \times 16_0) + (4 \times 16_{-1}) = (10 \times 1) + (4 \times 0.0625)$

Thus, $(A.4)_{16} = (10.25)_{10}$

7. Hexadecimal to Binary Number System Conversion

To convert from Hexadecimal to Binary:

- Each hexadecimal digit (0-9 and A-F) is represented by a 4-bit binary number.



Binary equivalent	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

- For each digit in the hexadecimal number, find its corresponding 4-bit binary equivalent and write them down sequentially.

Example: $(3A)_{16}$

- $(3)_{16} = (0011)_{2}$
- $(A)_{16} = (1010)_{2}$

Thus, $(3A)_{16} = (00111010)_{2}$

8. Binary to Hexadecimal Number System Conversion

To convert from Binary to Hexadecimal:

- Start from the rightmost bit and divide the binary number into groups of 4 bits each.
- If the number of bits isn't a multiple of 4, pad the leftmost group with leading zeros.
- Each 4-bit binary group corresponds to a single hexadecimal digit.
- Replace each 4-bit binary group with the corresponding hexadecimal digit.

Example: $(1111011011)_{2}$

0011 1101 1011
| | |
3 D B

Thus, $(001111011011\ )_{2} = (3DB)_{16}$

9. Binary to Octal Number System

To convert from binary to octal:

- Starting from the rightmost bit, divide the binary number into groups of 3 bits.
- If the number of bits is not a multiple of 3, add leading zeros to the leftmost group.
- Each 3-bit binary group corresponds to a single octal digit.
- The binary-to-octal conversion for each 3-bit group is as follows:

Octal	Binary Equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- Replace each 3-bit binary group with the corresponding octal digit.

Example: $(111101101)_2$

111 101 101

| | |

7 5 5

Thus, $(111101101)_2 = (755)_8$

10. Octal to Binary Number System Conversion

To convert from octal to binary:

- Each octal digit (0-7) corresponds to a 3-bit binary number.
- For each octal digit, replace it with its corresponding 3-bit binary equivalent.

Example: $(153)_8$

- Break the octal number into digits: 1, 5, 3
- Convert each digit to binary:
    - 1 in octal = 001 in binary
    - 5 in octal = 101 in binary
    - 3 in octal = 011 in binary

Thus, $(153)_8 = (001101011)_2$

Binary Arithmetic refers to performing arithmetic operations (like addition, subtraction, multiplication, and division) on numbers expressed in the **binary number system**.

The binary system uses only **two digits: 0 and 1**, unlike the decimal system which uses ten digits (0–9).

Key Points About Binary Arithmetic

1. **Number System Used:** Base-2 (digits are only 0 and 1).
2. **Common Operations:**
   - **Binary Addition**
   - **Binary Subtraction**
   - **Binary Multiplication**
   - **Binary Division**
3. **Carry/Borrow Rules:** Since the base is 2, carrying and borrowing happen when results exceed 1 or go below 0.

Binary Arithmetic Rules

1) Binary Addition Rules

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10 (i.e., 0 with a carry of 1)

2) Binary Subtraction Rules

- 0 – 0 = 0
- 1 – 0 = 1
- 1 – 1 = 0
- 0 – 1 = 1 (borrow 1 from the next higher bit)

3) Binary Multiplication Rules

- 0 × 0 = 0
- 0 × 1 = 0
- 1 × 0 = 0
- 1 × 1 = 1

4) Binary Division Rules

- 0 ÷ 1 = 0
- 1 ÷ 1 = 1
- Division by 0 is undefined.

Example: Binary Addition

```
   1011 (11 in decimal)
+  0101 (5 in decimal)
-----------
  10000 (16 in decimal)
```

Where Is Binary Arithmetic Used?

- Digital electronics and computers (CPU operations)
- Logic circuits and microprocessors
- Data processing and encoding