# Chapter 1

## Overview of Computer Software & Programming Language

### Introduction to Computer

The term "Computer" comes from the word "compute", which means, "to calculate". Hence, people usually consider to be a calculating device that can perform arithmetic operations at high speed.

A computer is an electronic device capable of performing commands, and these commands are generally inputs, outputs, storage, arithmetic and logical operations. More importantly it is a machine that is used to store, process and disseminate information. A computer program is actually a set of instructions inside a computer that is used to initiate and direct the intake, processing and dissemination of information inside the computer. The information and the instruction themselves are stored in the mass storage devices inside the computer. The data is entered through input devices such as keyboard, processed inside the CPU and routed to output devices (VDUs, printers, etc.). The processing of data takes place according to the prescribed set of instructions.

All the modern computers are digital and operate in binary system. In binary system we have only two digits (bits) - 0and 1 to represent data. Since the bit is a very small unit, virtually, it cannot represent any information while used single. A combination of eight bits is called a byte. A byte is the fundamental unit of data representation in computer. A byte can represent one character of data. Modern computers are capable of processing more than one byte at a time (2, 4 or 8 bytes).

*Key Points*

- Computer is basically a group of electronic devices that can accept data, conduct a series of arithmetic and logical operations on it and give result of these operations as information.
- Computer is the most versatile and efficient electronic device that a man has ever made.
- It is also termed as electronic data processor or information processor.
- Its main advantage lies in its enormously high speed, precision, reliability.
- Due to its versatility, easy to use, low cost etc. computers today have become extremely popular in all areas like scientific research, engineering, business, office automation, medical diagnosis, education, entertainment, Computer Aided Design (CAD) etc.

The computer consists of two major components - hardware and software:

### Hardware:

The physical components- from nuts and bolts to tiny chips to the processor connecting wires, and surrounding devices, which we can touch is called hardware. Some basic components of hardware are as follow:

- CPU (Central Processing Unit)
- Storage Device (Main & Secondary)
- Input Devices (Keyboard, Mouse, etc.)
- Output Devices (VDUs – Visual Display Unit, Printers, etc.)

## Software:

Software is any set of instructions that tells the hardware what to do. It is what guides the hardware and tells it how to accomplish each task. Without software, a computer is just an empty shell. Software can be placed into two separate categories system and application.

## Computer Software

Software, by definition, is the collection of computer programs, procedures and documentation that performs different tasks on a computer system. At the very basic level, computer software consists of a machine language that consists of groups of binary values, which specify processor instructions. Software is what gives computer a life. It is installed inside the computer in mass storage devices. Software can be placed into two separate categories system and application.

## System Software

System software is a program designed to control the Computer. It works remaining at the system level and is generally used to define the functioning of the hardware irrespective of its application area. The most popular designation of system software is operating system. An operating system is capable of handling all input devices (keyboard, mouse, disk, etc), output devices (screen, printer, disk, etc) and is able to manage the use of other resources such as memory, disk, CPU, etc. The examples of operating system software are DOS (Disk Operating System), Microsoft Windows, UNIX, LINUX, Apple Mac OS X etc.

| **Operating System:** Set of program that controls overall computer activity and provide services. |
|---|

Although operating system is important system software, the importance of other system software cannot be overlooked. Computer BIOS (Basic Input Output System) provide basic functions which help to operate and control the a computers system settings. Device firmware are system software used operate electronic gadgets and hardware built into a computer.

## Application Software

Application software is a program designed to perform specific task. It enables the end users to accomplish certain specific tasks. Business software, databases and educational software are some forms of application software. Different word processors, which are dedicated for specialized tasks to be performed by the user, are other examples of application software.

**Word processing software** allows you to process and manipulate text. For example, Microsoft Word, Word Perfect, etc. **Spreadsheets** allow mathematical calculations to be performed. MS Excel & Quattro Pro are some of its examples. Databases are more useful for organizing data. For example **database software** can keep track of your friend's name, address telephone numbers and you can add or remove whenever you want. Some examples of database software are MySQL, Oracle, MS-SQL, Paradox and MS Access. **Internet browsers** give us access to the world. When this software is installed you are connected to an ISP (Internet Service Provider), and you have access to the worldwide information. Common examples are Google Chrome, Mozilla Firefox, Netscape, Internet Explorer, etc. Similarly **image processing applications** such as Photoshop and image editor allows editing, resizing and manipulating images.

## History and Development of Computers

*Early Computing Machines:*

**Abacus:** Evolution of computer goes to a very elementary counting device called the ABACUS, developed around **5000 years ago in china**. It was the f**irst mathematical device** used to facilitate **arithmetic computation** and is still in use today, and may be considered the first computer. This device allows users to make computations using a **system of sliding beads arranged on a rack**. Early merchants used the abacus to keep trading transactions. But as the use of paper and pencil spread, particularly in Europe, the abacus lost its importance.

**Pascaline (Pascal Arithmetic Machine):** Pascaline was invented by **Blaise Pascal in 1642** that **added and subtracted**, automatically **carrying and borrowing digits from column to column**. Pascal built 50 copies of his machine, but most served as curiosities in parlors of the wealthy.

**Leibnitz Arithmetic Machine:** In 1694, a German mathematician and philosopher, Gottfried Wilhem von Leibnitz, **improved the pascaline** by creating a machine that could also **multiply and divide**. Like its predecessor, Leibnitz's mechanical multiplier worked by a **system of gears and dials**.

**Charles Babbage's Analytical and Difference Engine:** The real beginnings of computers started with the invention of Difference Engine in 1822 and an Analytical Engine by English Mathematics professor, Charles Babbage. The **Difference Engine** was a mechanical device that could **count and subtract**, but could run **only single algorithm**. The **Analytical Engine** on the other hand had advanced features and mainly contained **four components**: the store (*memory*), the mill (*computation unit*), the *input section* and the *output section*. The great advantage of the Analytical Engine was that it was general purpose. But, Charles Babbage's Engines suffered from the problem of frequent breakdowns. **Lady Augusta Ada** contributed in the **refinement of this machine** by inventing and using a new number system called the **binary number system** using only two digits '0' and '1' **instead of using decimal digits**.

**Hollerith Punch Card Machine:** In 1889, an American Inventer, Herman Hollerith invented a punch card machine **to process census data**. The data was in punch cards in coded form. He designed a wire brush with a wire for each possible location. The card was placed on a metal plate. Wires made contact with plate in the punched hole locations and generated electric pulses. This machine was **very successful for tabulation work**. This aided to the development of data processing machine. Hollerith brought his punch card reader into the business world, founding Tabulating Machine Company in 1896. **Hollerith's Tabulating Machine Company** eventually **merged with other companies in 1924** to become **International Business Machine (IBM)**.

In the ensuing years, several engineers made other significant advances like: Vannevar Bush developed a calculator for solving differential equations in 1931. The machine could solve complex differential equations. The machine was cumbersome because hundreds of gears and shafts were required to represent numbers and their various relationships to each other. To eliminate this bulkiness, John V. Atanasoff, a Professor at Lowa State College and his graduate student, Clifford Berry, envisioned an all-electronic computer that applied Booloean algebra to computer circuitry. This approach was based on the mid-19 th century work of George Boole who clarified the binary system of algebra, which stated that any mathematical equations could be stated simply as either true or false. By extending this concept to electronic circuits in the form of on or off, Atanasoff and Berry had developed the first all-electronic computer by 1940. Their project, however, lost its funding and their work was **overshadowed** by **similar developments by other scientists**.

In the 1930s American mathematician Howard Aiken developed the Mark I calculating machine, which was built by IBM. This electronic calculating machine used relays and electromagnetic components to replace mechanical components. In later machines, Aiken used vacuum tubes and solid state transistors to manipulate the binary numbers. Aiken also introduced computers to universities by establishing the first computer science program at Harvard University.

At the Institute for Advanced Study in Princeton, Hungarian-American mathematician **John von Neumann** developed one of the first computers used to solve problems in mathematics, meteorology, economics, and hydrodynamics. Von Neumann's 1945 **Electronic Discrete Variable Computer (EDVAC)** was the first electronic computer to use a program stored entirely within its memory.

**John Mauchely**, an Americian physicist, proposed an electronic digital computer, called the **Electronic Numerical Intergrator And Computer (ENIAC)**, which was built at the Moore School of Engineering at the University of Pennsylvania in Philadelphia by Mauchely and J. Presper Eckert, an American Engineer. ENIAC was completed in 1945 and is regarded as the first successful, general digital computer. It weighed more than **27,000 kg** and contained more than **18,000 vacuum tubes**. Many of **ENIAC's first task** were for **military purposes**, such as calculating ballistic firing tables and designing atomic weapons.

Eckert and Mauchley eventually formed their own company, which was then bought by the Rand Corporation. They produced the **Universal Automatic Computer (UNIVAC)**, which was used for a broader variety of commercial applications. UNIVAC was the first successful commercial computer.

In 1948, at Bell Telephone Laboratories, American physicists Walter Houser Brattain, John Bardeen, and William Bradford Shockely developed the transistor, a device that can act as an electric switch. The transistor has a tremendous impact on computer design, replacing costly, energy-inefficient, and unreliable vacuum tubes.

In the late 1960s integrated circuits(IC), tiny transistors and other electrical components arranged on a single chip of silicon, replaced individual transistors in computers. ICs became miniaturized, enabling more components to be designed into a single computer circuit. In the 1970s refinements in the integrated circuit technology led to the development of the modern microprocessor, integrated circuits that contained thousands of transistors. Modern microprocessors contain as many as 10 million transistors.

Manufacturers used integrated circuit technology to build smaller and cheaper computers. The first of these so-called personal computers (PCs) was sold by Instrumentation Telemetry Systems. The Altair 8800 appeared in 1975. It used an 8-bit Intel 8080 microprocessor, had 256 bytes of RAM. Refinement in the PC continued with the inclusion of video displays, better storage devices, and CPUs with more computational abilities. Graphical user interfaces were first designed by the Xerox Corporation, then later used successfully by the Apple Computer Corporation with its Macintosh computer. Today the development of sophisticated operating systems such as Windows 95,98,XP and UNIX enables computer users to run programs and manipulate data in ways that were unimaginable 50 years ago.

Possibly the largest single calculation was accomplished by physicists at IBM in 1995 solving one million trillion mathematical problems by continuously running 448 computers for two years to demonstrate the existence of a previously hypothetical subatomic particle called a **glueball**. Japan,

Italy, and the United States are collaborating to develop new supercomputers that will run these calculations one hundred times faster.

In 1996 **IBM challenged Gary Kasparov**, the world chess champion, to a chess match with a supercomputer called Deep Blue. The computer had the ability to compute more than 100 million chess positions per second. Kasparov won the match with **three wins, two draws, and one loss**. **Deep Blue** was the first computer to win a game against a world chess champion with regulation time controls. Deep Blue served as a prototype for future computers that will be required to solve complex problems.

## General Software Features and Recent Trends

Now a day software projects are becoming more and more complex in size, sophistication, and technologies used. Most of the software products are used by huge number of people, not only that, but also these software support various national languages and even come in different shapes and sizes like desktop, professional, standard, Enterprise Resource Planning (ERPs) packages and so on. Almost all application software products like word processors, ERP packages, etc support more than one hardware and/or software platform. For example- we have many web browsers for the PC and Mac; we have database management systems (DBMS) that run on MVS, UNIX, Linux, Windows NT, and so on. The advancements and the competition in technology are driving software vendors to include new features and additional functionality to their products- just to stay in business. The Information Technology (IT) is revolutionizing the way we live and work. The digital technology has given mankind the ability to treat information with mathematical precision, to transmit it at very high accuracy and to manipulate it at will.

## Features:

The trends that play a vital role in the development of software products are described as below:

- **Ease of use:** The software systems applications are becoming more and more easy to use. System analysts and software developers are concerned on ensuring that the software they develop are user-friendly than that of their competitor's products. Hence, the user interfaces are more intuitive, there is context sensitive help, the error messages are more descriptive, there are wizards and templates to help the user when they encounter a problem.
- **Graphical User Interface (GUI):** Today's software applications and products provide users with graphical, intuitive and easy-to use interfaces. Nowadays the users don't have to remember the shortcut keys or cryptic system commands that were a must in the character based era. Today, almost any tasks can be accomplished just by a mouse click. For example- In a DOS environment, to copy a file, one needs to know the command for copying files, its exact syntax and so on, whereas in case of the Windows environment, we just have to drag the files that we want to copy from the source to destination.
- **Requirement of more powerful hardware:** As software vendors are incorporating more and more new features into their products, the software needs more powerful machines to run. Thus, they need more main memory, more secondary storage, and faster and powerful processors. Even it is not very difficult to go for new powerful computers today as the price of computers are decreasing day by day.
- **Multi-platform capability:** Software applications today are not developed for just one platform. Most software applications support multiple platforms, i.e. both hardware and software platforms. There are software applications which support hardware platforms ranging from mainframes to PCs and various software platforms like MVS, Windows, AIX, UNIX, Solaris & so on. Database like IBM's DB2 Universal is available for a variety of hardware and

software platforms. Today's software application supports multiple currencies and multiple languages which are it's another important feature. Many software vendors are providing their application in different languages like English, Arabic, Japanese, Chinese and so on.

- **Network Capabilities:** Nowadays, network computers are becoming popular as they can work with minimal memory, processor power and disk storage. Network computers are connected to a network, especially the Internet. The main idea behind network computers is that many users who are connected to a network don't need all the computer power they get from a typical personal computer (PC). Instead, they can rely on the power of the network servers. As the popularity of network computers is increasing, the demands for software that can run these computers are also increasing and hence the software applications of today and tomorrow will have such capability.
- **Compatibility with other software:** Most of the software products are compatible with each other nowadays. For example, HTML documents and other text documents can be imported into a MS-Word document. As newer versions of software are released, most software vendors maintain backward compatibility i.e. compatibility with previous versions. These two features– backward compatibility and compatibility with other products make it easier for the users, as they can choose the application they want and still use the old files they created by using other applications or using earlier versions of the same application.
- **Object Linking and Embedding:** We also have the mechanism to interact with other software systems. One of the methods to integrate external tolls into an application is by using the Object Linking and Embedding (OLE) architecture to embed or link a component from another application running on the computer. OLE is a compound document standard developed by Microsoft Corporation. It enables us to create objects with one application and then embed or link them into a second application. This linking and embedding of components enable applications to share the components.
- **Group-work capabilities:** Technologies, which support collaboration, are in greater demand today than ever before. So, vendors are integrating collaborating technologies into their products. Distributed workforces, getting products to market as quickly as possible and information overload are just a few of the motivational aspects pushing collaboration technology development.
- **Mail Enabling:** The process through which email is gradually replacing many of the single-purpose applications now used on PCs is known as the mail enabling of an application. In its simplest form, a mail-enabled application is a Windows program that has a 'Send' command in its 'File' menu. For example, the Microsoft Word 2000 has a 'Send To' menu item in the 'File' menu where one can choose to send the document to a mail recipient, fax recipient and so on.
- **Web Enabling:** With the ever-increasing popularity of Internet and the amount of information that is available on the net, most of the software applications are now web-enabled. Web enabling helps the user in many different ways. During installation, most of the applications will automatically connect the Internet to the vendor's web site and will register their products (previously one had to fill in a paper form and mail or fax it to the vendor).

## Generation of Computer

Each generation of computer is characterized by major technological development that fundamentally changed the way computers operate, resulting in increasingly smaller, cheaper, more powerful, more efficient, and reliable devices. According to this there are five generation of computers as described below:

1. **The First Generation: (1945-55)** *Vacuum Tubes and Plug Boards*

*Characteristics:*

- These machines were enormous, filling up entire rooms with tens of thousands of vacuum tubes, but were much slower than even the cheapest personal computer available today.
- First generation computer relied on machine language to perform operations, and they could only solve one problem at a time.
- Input was punched on punched cards and paper tape, and output was displayed on printers. Some examples are UNIVAC, ENIVAC etc.

*Advantages:*

- Vacuum tubes were the only electronic components available during those days
- Vacuum tube technology made possible the advent of electronic digital computers
- These computers were the fastest calculating device of their time. They could perform computations in milliseconds.

*Disadvantages:*

- Too bulky in size so not portable
- Unreliable
- Thousands of vacuum tubes that were used emitted large amount of heat and burnt out frequently
- Air conditioning required
- Prone to frequent hardware failures
- Constant maintenance required
- Manual assembly of individual components into functioning unit required
- Commercial production was difficult and costly so limited to commercial use

2. **The Second Generation: (1955-65)** *Transistors and Batch Systems*

*Characteristics:*

- Transistors were used as a basic building block for second generation computers.
- Second generation computers moved from cryptic binary machine language to symbolic, or assembly language, which allows programmers to specify instructions in words. High level languages were also being developed at this time like early version of FORTRAN and COBOL.
- These were also the first computer that stored their instructions in their memory.
- Second generation computer still relied on punched card for input and printer for output.

*Advantage:*

- Smaller in size as compared to first generation computers
- More reliable

- Less heat generated
- These computers were able to reduce computational times from milliseconds to microseconds
- Less prone to hardware failures
- Better portability
- Wider commercial use

*Disadvantage:*

- Air conditioning required
- Frequent maintenance required
- Manual assembly of individual components into a functional unit was required
- Commercial production was difficult and costly

Advances in electronics technology continued and the advent of "microelectronic" Technology made it possible to integrate large number of elements into very small (less than 5mm square) surface of silicon known as " chips". This new technology was called "integrated circuit" (ICs).

3. **The Third Generation: (1965- 1980)** *ICs and Multiprogramming*

Computer chips called IC (Integrated Circuits) were introduced, which was a basic building block for this generation of computer. Concept of Multiprogramming was introduced in which when no. of jobs are to be performed, main memory is divided into several pieces with different job in each partition. While one job was waiting for input/output to computer, another job could be using the CPU. If enough job could be held in main memory at once, the CPU could be kept busy nearly 100% of the time. Another major feature present in the 3rd generation operating system was

**Spooling:** (Simultaneous peripheral operation on line) Is the ability to read the jobs from cards on to the disk as soon as they were bought to the computer room? Then, whenever a running job finished, the operating system could load a new job from the disk into the partition & run it. This technique is called spooling .And was used for output.

**Time Shared System:** Sharing one processor by several no. of computer. If 20 users are logged in and 17 of them are thinking or talking or drinking coffee , the CPU can be allocated in turn to the three jobs that want service. Since people debugging programs usually issue short commands rather than long ones, the computer can provide fast , interactive service to a no. of users & also work on big batch jobs in the background when the CPU in other side is idle. Instead of punched card and printout, users interact with third generation computers through keyboard and monitors.

*Advantages:*

- Smaller in size as compared to previous generation computer.
- Even more reliable than second generation
- Even lower heat generated than 2nd generation
- These computers were able to reduce computational times from microsecond to nanoseconds.
- Maintenance cost is low because hardware failures are rare.
- Easily portable
- Totally general purpose. Widely used for various commercial applications all over the world
- Less power requirement than previous generation

- Manual assembly of individual components into a functioning unit not required. So human labor and cost involved at assembly stage reduced drastically.
- Commercial production was easier and cheaper

*Disadvantages:*

- Air-conditioning required in many cases
- Highly sophisticated technology required for the manufacture of IC chips.

4. **Fourth Generation (1980 onwards)**

Initially, the integrated circuits contained only about ten to twenty components. This technology was named small scale integration (SSI). Later, with the advancement in technology for manufacturing IC's, it became possible to integrate up to a hundred components on a single chip. This technology came to be known as medium scale integration (MSI). Then came the era of large scale integration (LSI), when it was possible to integrate over 30,000 components onto a single chip. Then more advance technology like very large scale integration (VLSI) came. Development of GUI, mouse, and handheld devices took place in this generation.

*Advantages:*

- Smallest in size because of high component density
- Very reliable
- Heat generated is negligible
- No air conditioning required in most cases
- Much faster in computation than previous generations
- Hardware failure is negligible and hence minimal maintenance is required
- Easily portable because of their small size
- Totally general purpose
- Minimal labor and cost involved at assembly stage
- Cheapest among all generations

Disadvantages:

- Highly sophisticated technology required for the manufacture of LSI chips.

5. **Fifth Generation (Present and Beyond)**

Defining the fifth generation of computers is somewhat difficult because the field is in its infancy. Using recent engineering advances, computers are able to accept spoken word instructions (voice recognition) and imitate human reasoning. The ability to translate a foreign language is also moderately possible with fifth generation computers.

Many advances in the science of computer design and technology are coming together to enable the creation of fifth-generation computers. Some of such engineering advances are **parallel processing**, and another is **superconductor technology**, which allows the flow of electricity with little or no resistance, greatly improving the speed of information flow. The concept of using **bio-chips** in computers is also being worked out in the research labs to make the computer to have an **ability to think**.

## Types of Computers

According to the type of data they handle, the computers can be classified into two categories: Analog computer and Digital computer

**Analog Computer**

- The analog computers operate on continuous-time basis and process the continuous time information i.e. analog computer takes input continuously (in analog form).
- The input variables of this type of computer are acceleration, velocity, displacement etc.
- Cheaper and easier to use
- Less accurate and limited storage capacity
- Basic building block of analog computer is an operational amplifier, which is an electronic device that can amplify the electrical signals.
- The application of analog computer is in solving the differential equations. Its basic purpose is to predict the behavior of a physical system that can be described by a set of algebric or differential equations.

**Digital Computer**

Digital computer is an electronic device made of simple logic gates and memory devices organized to perform complex calculations at high speed by automatically processing information in the form of electrical pulses.

- They take input in the form of discrete electronic pulses.
- They are capable of storing data for processing, editing, printing etc.
- Basic building blocks of digital computers are: input devices, CPU, output devices, and memory.

General purpose computers come in many sizes and with varying capabilities. The term that describes the different types of computers are categorized as follows:

- Supercomputer
- Mainframe computer
- Minicomputer
- Microcomputer or Personal Computer

**Supercomputer**

Supercomputers are the most powerful, largest and fastest computers where numerical computations are carried out at the speed of 50 million operations per second. They are built to process huge data. Used in specialized area like defenses, aircraft design, computer generated movies, weather research but not used in commercial data processing. The first supercomputer was the ILLIAC IV. Modern supercomputer posses several independent processors which cooperate in the execution of single program in the fashion of parallelism (decomposition of program into components, known as task or threads). Some examples of supercomputers are: Intel ASCI Red-1997 supporting 9216 processors and 584 MB memory, Cray SVI introduced in 1998 supporting 1024 microprocessors and 1024 GB memory.

**Mainframe computer**

The largest type of computer in common use is the mainframe computer. It is less powerful and cheaper as compared to supercomputers. These computers are used where many people in large organizations

like scientific and business applications need frequent access to same information. Speed of mainframe computers are several MIPS and support more than 1000 terminals. It has large memory, on line secondary storage capacity, high speed cache memory which makes it faster than micro and minicomputer. It supports for multiprogramming and time sharing. It sizes from small to very large.

**Minicomputer**

Minicomputers are the computers that lie somewhere between mainframes and personal computers. Like mainframes, minicomputers can handle a great deal more input and output than personal computers can. Although minicomputers are designed for a single user, many of minicomputers can handle dozens or even hundreds of terminals. Minicomputers supports distributed data processing system, so, large numbers of minicomputers can replace mainframe computers. Examples of minicomputers are DEC, PDP-11 etc.

**Microcomputer (Personal Computer)**

When we use the term personal computer or microcomputer we mean small computers that are commonly found in offices, homes, colleges etc. Personal computers come in many shapes, sizes and brands. Although most models sit on desktops, others stand on floor, and some are portable. A personal computer is a full-fledged computer system which uses microprocessor as its CPU. It evolved with microchip in 1970s. It has internal memory in the form of RAM and ROM. The first personal computer was IBM PC in 1981. Microcomputer supports multi-user systems and now a days are used especially in network.

## Programming Languages

Programming languages are the basic building blocks for all software, allowing people to tell computers what to do and the means by which systems are developed. A programming language or computer language is a standardized communication technique for expressing instructions to a computer. It is a set of syntactic and semantic rules used to define computer programs. A language enables a programmer to precisely specify what data a computer will act upon, how these data will be stored / transmitted, and precisely what actions to take under various circumstances. It is appropriately called language since it resembles the characteristics of the human language. Sometimes these languages are not readily understood by the computers and need to be converted to machine understandable form by some conversion process.

Programming languages are generally classified in two categories; they are low level language & high level language.

**Low Level Language:**

A low level language is a programming language that is much closer to the hardware. It requires a thorough knowledge of the hardware for which the program is being created. It can be divided into two types, Machine language and assembly language.

**1. Machine Language:**

The lowest level language the computer understands. The computer converts all high-level languages into machine language before executing any statements. Machine language is written in binary form that a computer can execute directly. These are also known as machine code or object code. This language not at all feasible for humans to use as it consists entirely of binary numbers. If the word length of the machine is 16 bits (two byte) and we have to add two numbers (say 2 and 4) then we have to remember binary number that is used for addition (say 0100000011001100). This language is fastest in terms of execution in the sense that it does not need to be translated.

**2. Assembly language:**

If one would be able to remember binary numbers then it would be much easier to program. However binary numbers are hard to remember and it is tedious to program using it. In assembly language the binary bits are replaced by certain keywords which we call mnemonic. These mnemonics are in fact abbreviated letters which are much easier to remember. In the above example, if 0100000011001100 is instruction used to add two numbers, then ADD would be the easiest mnemonic to remember for it. This language is easier to learn and more portable than machine language. An assembler is used to convert assembly language into machine language. It is still faster as compared to high level language and slower than lowest level language (Machine language).

| In an assembly language, mnemonics are used to represent operations to be performed by the computer. |
|---|

**High Level Language:**

High level language is much closer to human language so it is more suitable to write code in high level language. It is more or less independent of the particular type of computer used (i.e. more portable) and has its own set of rules called syntax. Its main advantage is that it is easier to read, write and maintain. The first high level programming languages were designed in the 1950s. Now there are dozens of such languages available such as BASIC, COBOL, C, C++, FORTRAN, LISP, PASCAL, prolog, etc. This language is lot slower in terms of execution as compared to low level language.

A programmer is further away from details inside the computer while writing a program in high level language. In other words there is an abstraction from details of the low level machine language. The amount of abstraction determines how high the language is. A high level language can be further be categorized into object oriented and procedural programming language.

**1. Object-Oriented Programming Language**

In object oriented programming; the data (i.e. variables) and code are combined to form objects. This allows more effective code duplication which is not the case when programs are divided into subroutines (or functions). The required programming parts can be called again and again within the program. Its main distinction with procedural programming is that in procedural programming task is divided into subroutines, structures and functions. Whereas in object oriented programming, data as well as functions and subroutines are encapsulated to form objects.

**2. Procedural Programming Language**

Procedural programming is based upon the idea of series of procedure calls. A programmer programming in this language can exactly specify a sequence of steps in order to perform a particular task. A procedure may be a program in itself that may be called within a main program, a subroutine or another program. A programmer knows exactly what is to be accomplished at the end of the program and uses a sequence of algorithmic steps in order to achieve it. C- Language is more often referred to as procedural programming language as opposed to procedural programming language such as C++ which is object oriented.

> Languages which express step-by-step algorithms written to solve a problem are known as procedural languages whereas those which express specifications of a program to be solved are known as non-procedural. Procedural languages are designed using a set of syntax rules, which precisely specify the 'words' of the language, and how they may be combined legally.

## Other Categorization of High Level Languages

**1. Problem oriented language**

It is a computer language that is designed to handle a particular class of problem. Problem oriented language can be classified as follows:

I. Numerical Problem Oriented – MATLAB

II. Symbolic Problem Oriented – MATHMATICA

III. Publishing Problem Oriented – LATEX

**2. Non – Procedural Language**

In Non – Procedural language problem is solved by expressing its specification rather than doing step-by-step in procedural language. These can be further classified as:

I. Functional – LISP

II. Logical – ML

**3. Scripting Language**

In Scripting languages assume that a collection of useful programs, each performing a task, already exists in other languages. It has facilities to combine these components to perform a complex task. For e.g. visual basic, Perl, etc.

**4. Constructed language**

It is a formal natural language that brings fiction or other works to life. More importantly these languages are associated more with formal communication rather than computing application.

**5. Natural language**

It is a language that arises automatically in unpremeditated fashion from human language. It may be spoken, written or formal and is differentiated from constructed languages in the aspect of mathematical logic. Bats communicating in infra sound are all examples of natural language.

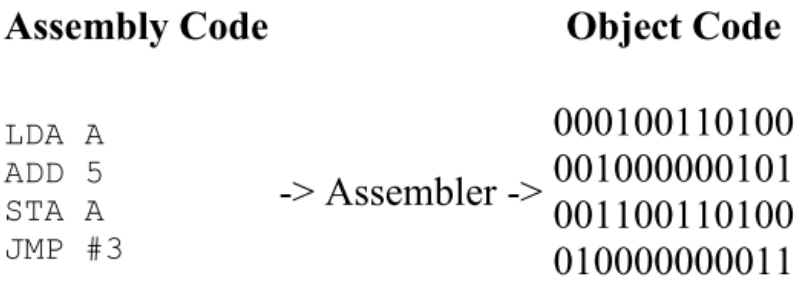
## Generation of Programming Languages

Programming languages are classified into five generations depending upon the power and flexibility of programming styles.

**First Generation Programming Language (1GL)** - There was no strict demarcation between first and second generation languages in early days of its creation. In fact the terms were not used until third generation programming language emerged. The first generation programming language consisted entirely of binary numbers which were entered through the front panel switches of the computer system; thus, they are appropriately named as machine level language.

The main and perhaps the only benefit in writing machine level language is that the code written by the user does not require any translation and hence can run very fast and efficiently. This language is far more difficult to learn than other generational programming languages and it is equally difficult to trace errors, if any were to occur. Instructions (programs) are stored in memory locations sequentially and therefore entire code needs to be changed if insertion or deletion in the code needs to be made. Furthermore, this generation of programming language is significantly less portable than other generation of languages.

**Second Generation Programming language (2GL)** - Assembly level languages are suitably included in second generation of programming language since their creation immediately followed the emergence of first generation programming language. The tedious coding consisting entirely of numbers is somewhat eliminated in this language, since certain mnemonics are used in place of numbers. However, translation is required to convert these mnemonics into machine code. Therefore the coding process is slower than machine language. The language is specific to particular processor or family thus less portable than other higher generational languages. Although primitive, this language is still widely used to program microcontrollers, device drivers, etc. (LDA – load, STA – store, JMP – jump)




**Third Generation Programming Language (3GL)** - As a refined version of second generation language, the third generation language emerged which brought a structured logic to the software, making programming more user friendly. The high level languages of this generation are more English-like in structure where each and every statement is taken from natural language. The codes written in this language need to be translated, which is done by a compiler or interpreter. Due to this translation process the codes run much slower. The codes are easier to edit and debug and are machine independent. A programmer does not require the hardware and details of the machine being programmed. Examples of 3GL are C, C++, C#, PASCAL, FORTRAN, JAVA, etc.

**Fourth Generation Programming Language (4GL)** - The fourth generation language is one step ahead of the third generation language. There are fewer options for the programmers, but the programs are much easier to write and debug than 3GLs. There built in GUI (Graphical user interfaces) objects like buttons, dropdown menus, add-ins, etc and no separate code needs to be written for them. These languages are particularly developed with the viewpoint of solving a particular

class of problem. Given the right kind of problem, the use of an appropriate 4GL can be spectacularly successful. All 4GLs are designed to reduce programming effort, the time it takes to develop software, and the cost of software development. Compilers and interpreters are still needed to convert it into machine code. Thus a single line code in this language is equivalent to more lines of coding in lower level language. Therefore the programs generated using these languages are larger and much slower. For e.g. SQL, ORACLE Repots, MATLAB, ColdFusion, CSS etc.

**Fifth Generation Programming Language (5GL) -** The fifth generation programming language uses human's natural language as program source code. It is designed to make the computer solve the problem without the programmer. The programmer only needs to worry about what problems need to be solved and what conditions need to be met, without worrying about how to implement a routine or algorithm to solve them.

Human languages by nature are full of ambiguity. A single statement in natural language is intended to provide a particular meaning can serve to present an entirely new meaning altogether. But this ambiguity cannot be handled by a computer which requires specific and well defined instructions. Therefore the use of 5GL has not become a reality yet and is still in the research phase. 5GLs are mostly used in artificial intelligence research. Prolog, OPS5, mercury, etc. are examples of fifth generation languages.

## Compiler

To translate language spoken in one country to a different type of language spoken in another country, we need a translator. Similarly to translate the language more closely to human (high level language) to a language understood by computers (machine level language) we need some translator. This type of translator is known as a compiler.

## Interpreter

An interpreter is another type of translator which translates high level language into machine level language. An interpreter however works differently as compared to a compiler. It analyzes and translates each line of source code in succession without looking at the entire program. In fact each program is executed letter by letter, word by word and then line by line. Hence it is an incredibly slow process. It does not generate executable program as in the case of compiler. The advantage of interpreters is they can execute a program immediately whereas compilers require some time before an executable program emerges. However, programs produced by compilers run much faster than same program executed by an interpreter. Compiled programs generally run faster than interpreted programs. The advantage of an interpreter, however, is that it does not need to go through the compilation stage during which the machine instructions are generated. This program can be time-consuming if the program is long. The interpreter on the other hand can immediately execute high level programs. For this reason, the interpreters are sometimes used during the development of a program, when a programmer wants to add small sections at a time and test them quickly.