Programming languages and Problem Solving

Chapter 1: Introduction

History of Computing and Computers

- The history of Computers starts about 2000 years ago in Babylonia, at the birth of Abacus, a wooden rack holding two horizontal wires with beads strung on them.
- Blaise Pascal builded the concept of a first digital computer in 1642 that added numbers entered with the dials which was to help his father, a tax collector.
- There was the development of the punch cards in 1890 by Herman Hollerith. This device could read the information that had been punched into the cards automatically without any human help. This reduced the reading errors dramatically which then increased the workflow
- IBM improved the development of the punch card and used it in electromechanical devices in which electrical power provided mechanical motion

History of Computing and Computers

- The start of a world war II produced a large need for computer capacity. In 1942 John P. Eckert from Electrical Engineering of University of Pennsylvania decided to build a high speed electronic computer named ENIAC (Electrical Numerical Integrator and Calculator). It used 18000 vacuum tubes, about 1800 feet square feet of floor space and consumed about 180,000 watts of electrical power. It was about 1000 times faster than the previous generation of relay computers.
- In the early 50s, The discovery of the Transistor Circuit changed the landscape of the electronic - computer field. By 1960s this was upgraded such that RAM capacities increased from 8000 to 64000 words with access time (2 to 3 ms). These machines were expensive and they were mostly operated by large industry or laboratories and government
- Many companies such as Apple Computer and Radio Shack introduced a successful PC’s in 1970’s. By 1980’s the manufacturing of semiconductor were mostly led by Intel and Motorola Corporation. With this advancement by late 1980’s some personal computer were run on microprocessor that handled 32 bits of data at a time that could process about 4,000,000 instruction per second

Generation of Computer

A generation refers to the state of improvement in the development of a product. This term is also used in the advancement of the computer technology. With each new generation, the circuitry has gotten smaller and more advanced than the previous generation from it. As a result speed, power and memory of computer has proportionally increased.

First Generation of Computer

The period of first generation was from 1940-1956.

The computers of first generation used vacuum tubes as the basic components for memory and circuitry for CPU (Central Processing Unit).

These tubes, like electric bulbs, produced a lot of heat and the installations used to fuse frequently.

Therefore, they were very expensive and• Therefore, they were very expensive and only large organizations were able to afford it.

In this generation, mainly batch processing operating system was used.

Punch cards, paper tape, and magnetic tape was used as input and output devices.

The computers in this generation used machine code as the programming language

The main features of the first generation are:

- Vacuum tube technology
- Unreliable
- Supported machine language only
- Very costly
- Generates lot of heat
- Slow input and output devices
- Huge size
- Need of AC• Need of AC
- Non-portable
- Consumes lot of electricity

Some computers of this generation were:

- ENIAC
- EDVAC
- UNIVAC
- IBM-701
- IBM-750

Second Generation of Computer (1956-1963)

The replacement of vacuum tubes by transistors saw the advent of the second generation of computing.

• Although first invented in 1947, transistors weren't used significantly in computers until the end of the 1950s.

• They were a big improvement over the vacuum tube, despite still subjecting computers to damaging levels of heat.

• However they were hugely superior to the vacuum tubes, making computers smaller, faster, cheaper and less heavy on electricity use.

• They still relied on punched card for input/printouts.

• The language evolved from cryptic binary language to symbolic ('assembly') languages. This meant programmers could create instructions in words. About the same time high level programming languages were being developed (early versions of COBOL and FORTRAN).

• Transistor-driven machines were the first computers to store instructions into their memories – moving from magnetic drum to magnetic core 'technology'. The early versions of these machines were developed for the atomic energy industry.

The main features of second generation are:

- Use of transistors
- Reliable in comparison to first generation computers
- Smaller size as compared to first generation computers
- Generates less heat as compared to first generation computers
- Consumed less electricity as compared to first generation computers
- Faster than first generation computers
- Still very costly
- AC required• AC required
- Supported machine and assembly languages

- Some computers of this generation were:
  - IBM 1620
  - IBM 7094
  - CDC 1604
  - CDC 3600
  - UNIVAC 1108

Third Generation Computers(1964-71)

By this phase, transistors were now being miniaturised and put on silicon chips (called semiconductors).

- This led to a massive increase in speed and efficiency of these machines.
- These were the first computers where users interacted using keyboards and monitors which interfaced with a monitors which interfaced with an operating system, a significant leap up from the punch cards and printouts.
- This enabled these machines to run several applications at once using a central program which functioned to monitor memory.
- As a result of these advances which again made machines cheaper and smaller, a new mass market of users emerged during the ‘60s.

The main features of third generation are:

- IC used
- More reliable in comparison to previous two generations
- Smaller size
- Generated less heat
- Faster
- Lesser maintenance
- Costly
- AC required
- Consumed lesser electricity• Consumed lesser electricity
- Supported high-level language

• Some computers of this generation were:

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

Fourth Generation Computers (1971-Present)

This revolution can be summed in one word: Intel. The chip-maker developed the Intel 4004 chip in 1971, which positioned all computer components (CPU, memory, input/output controls) onto a single chip.

- What filled a room in the 1940s now fit in the palm of the hand.
- The Intel chip housed thousands of integrated circuits.
- The year 1981 saw the first ever computer (IBM) specifically designed for home use and 1984 saw the MacIntosh introduced by Apple.
- Microprocessors even moved beyond the realm of computers and into an increasing number of everyday products.
- The increased power of these small computers meant they could be linked, creating networks.
- Which ultimately led to the development, birth and rapid evolution of the Internet.
- Other major advances during this period have been the Graphical user interface (GUI), the mouse and more recently the astounding advances in lap-top capability and hand-held devices.

The main features of fourth generation are:

- VLSI technology used
- Very cheap
- Portable and reliable
- Use of PCs
- Very small size
- Pipeline processing
- No AC required
- Concept of internet was introduced
- Great developments in the fields of networks
- Computers became easily available

- Some computers of this generation were:
  - DEC 10
  - STAR 1000
  - PDP 11
  - CRAY-1(Super Computer)
  - CRAY-X-MP(Super Computer)

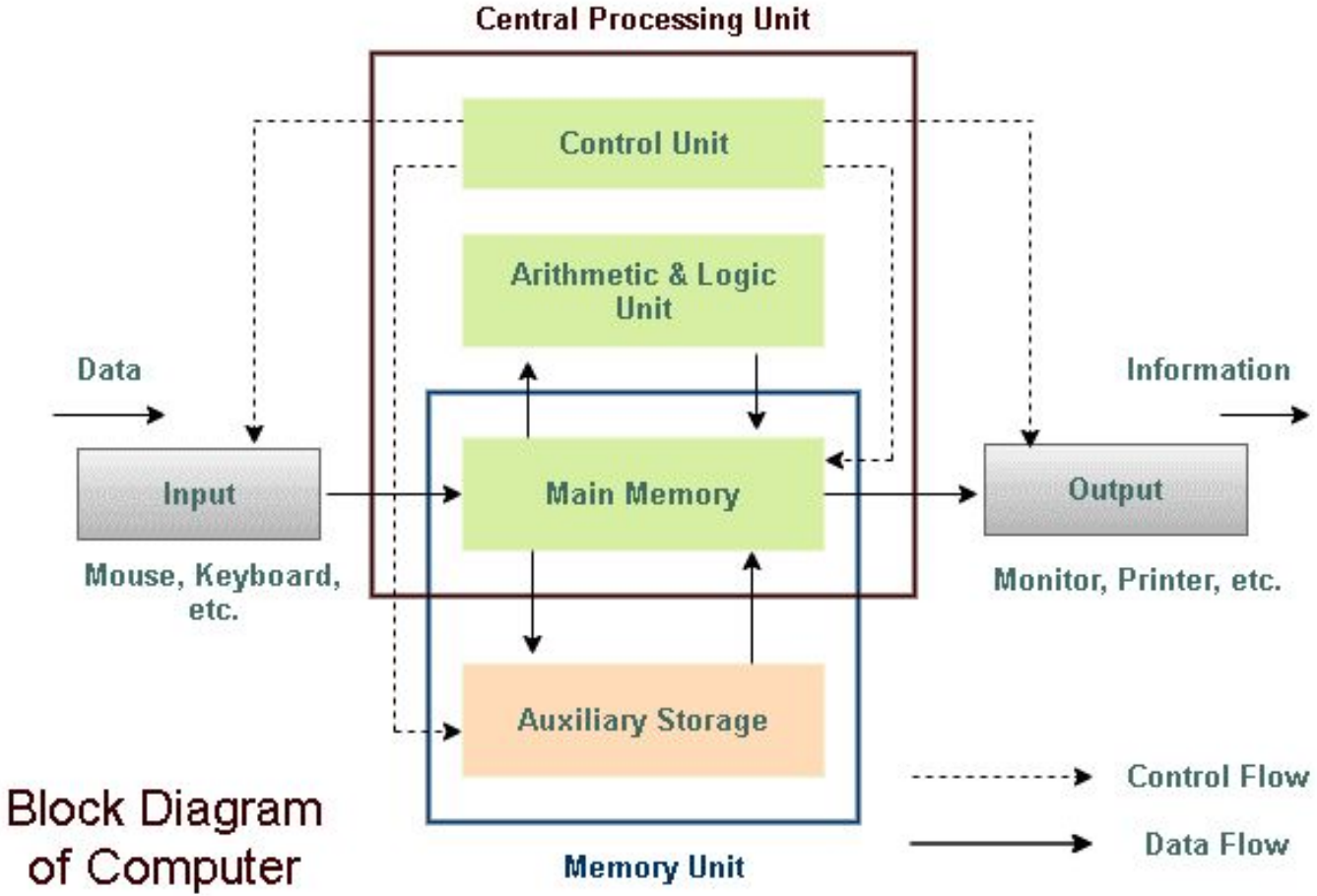
Fifth Generation Computers (Present and beyond)

Computer devices with artificial intelligence are still in development, but some of these technologies are beginning to emerge and be used such as voice recognition.

• AI is a reality made possible by using parallel processing and superconductors. Leaning to the future, computers will be radically transformed again future, computers will be radically transformed again by quantum computation, molecular and nano technology.

• The essence of fifth generation will be using these technologies to ultimately create machines which can process and respond to natural language, and have capability to learn and organise themselves.

Block Diagram of a Computer



Input Unit: The process of sending the data and instructions for processing through some suitable devices like Keyboard, Mouse, scanner etc. is called Input.

The device translate the data from human understandable form into electronic impulses which are understood by the computer.

Storage Unit: The storage unit of the computer holds data and instructions that are entered through an input unit before they are processed. It preserves the intermediate and final results before these are sent to the output devices. It also saves data for later use. It is generally divided into two categories.

a. Primary Storage: Stores and provides very fast. This memory is generally used to hold the program being currently executed in the computer, the data being received from the input, the intermediate and final result of the program. The primary memory is temporary in nature. The data is lost when the computer is switched off. In order to store it permanently, data has to be moved to secondary memory. Thus most computer have limited primary storage capacity.
b. Secondary Storage: It stores programs, documents, data permanently. The programs that you run on the computer are first transferred to primary memory and the results are saved in the secondary memory. For eg: Hard Disk

CPU (Central Processing Unit)

Central Processing Unit or the CPU, is the brain of the computer. It works the same way a human brain works. As the brain controls all human activities, the CPU too controls all tasks.

Moreover, the CPU conducts all the arithmetical and logical operations in the computer.

Now the CPU comprises of two units, namely – ALU (Arithmetic Logic Unit) and CU (Control Unit). Both of these units work in sync. The CPU processes the data as a whole.

ALU

The Arithmetic Logic Unit is made of two terms, arithmetic and logic. There are two major functions that this unit performs.

1. Data inserted through the input unit into the primary memory. Performs the basic arithmetical operation on it. Like addition, subtraction, multiplication, and division. It performs all sorts of calculations required on the data. Then sends back data to the storage.
2. The unit is also responsible for performing logical operations like, AND, OR, Equal to, Less than, etc. In addition to this it conducts merging, sorting, and selection of the given data.

CU (Control Unit)

The control unit as the name suggests is the controller of all the activities/tasks and operations. All this is performed inside the computer.

The memory unit sends a set of instructions to the control unit. Then the control unit in turn converts those instructions. After that these instructions are converted to control signals.

These control signals help in prioritizing and scheduling the activities. Thus, the control unit coordinates the tasks inside the computer in sync with the input and output units.

Output

All the information sent to the computer once processed is received by the user through the output unit. Devices like printers, monitors, projector, etc. all come under the output unit.

The output unit displays the data either in the form of a soft copy or hard copy. The printer is for the hard copy. The monitor is for the display. The output unit accepts the data in binary form from the computer. It then converts it into a readable form for the user.

Classification of Computers

On the basis of working mode

- Analog Computer
- Digital Computer
- Hybrid Computer

On the basis of size

Analog Computer

Analog Computers are designed to process the analog data. Analog data is continuous data that changes continuously and cannot have discrete values such as speed, temperature, pressure and current.

The analog computers measure the continuous changes in physical quantity and generally render output as a reading on a dial or scale.

Analog computers directly accept the data from the measuring device without first converting it into numbers and codes.

Speedometer and mercury thermometer are examples of analog computers.

Digital Computer

Digital Computer is designed to perform calculations and logical operations at high speed.

It accepts the raw data as digits or numbers and processes it with programs stored in its memory to produce output.

All modern computers like laptops and desktops that we use at home or office are digital computers.

Hybrid Computer

Hybrid Computer has features of both analog and digital computer.

It is fast like analog computer and has memory and accuracy like digital computers. It can process both continuous and discrete data.

So it is widely used in specialized applications where both analog and digital data is processed.

For example, a processor is used in petrol pumps that converts the measurements of fuel flow into quantity and price.

Micro Computers

These ae also known as personal computers (PCs). They are small in sizes and relatively inexpensive.

They are highly flexible and designed for small scale personal use and business application. these are based on microprocessor technology, a silicon chip containing the important circuitry to perform operations.

IBM, Dell, Sony, HP and Apple are the manufacturers of microcomputers. Nowadays they have come more advance and powerful in technology.

The business community uses personal computer for word processing, desktop publication, inventory and accounting software like QuickBooks and spreadsheets. At home users perform surfing internet and playing games.

Main Frame Computer

Mainframe computers can store huge amount of data. They are large in size and memory, power and expensive.

These machines are capable to different operating system on the same time and support time sharing and virtual memory.

Universities, banking institutions, stock exchanges, insurance companies are some examples of using mainframe servers for maintaining students' data, ATM transactions, financial transactions, clients and employees' records.

The main difference between supercomputer and mainframes is that supercomputer has a speed of calculations and mostly used for space scientific and engineering related problems i.e. data crunching while mainframe servers are used for transaction process.

Super Computer

The term “super computer” is considered the most powerful computers in performance and processing. Supercomputers are capable to solve immense mathematical calculations, scientific problems and hence it is called number crunchers. They are difficult to design and need teamwork research and development.

Theses monster machines are capable to play their role in the field of weather forecasting, quantum mechanics and physical simulation (spaceship aerodynamics and testing of nuclear weapons.

Another important application of supercomputer is oil exploration. A program can simulate areas where oil is likely to be found and can predict costs and methods of findings and retrieving oil.

Mini Computer

Compare to mainframe these are midrange computers, small in size and more compact and less expensive.

They are designed for computerization of data, scientific research, industrial process, data collection and telephone switches.

Minicomputers are multiprocessing system and can support upto 200 user at a time but the power of processing is not as great as the mainframe and supercomputers.

Timesharing, batch processing and online processing is available on these midrange computers.

Computer Software

Computer software is programming code executed on a computer processor. The code can be machine-level code, or code written for an operating system.

An **operating system** is software intended to provide a predictable and dependable layer for other programmers to build other software on, which are known as **applications**. Operating systems can be found on all smartphones, tablets, and desktop computers. These systems give the device the functionality it needs.

For example, Apple's iOS and Android's OS were developed specifically for portable devices.

Computer software provides a dependable layer for hardware manufacturers. This standardization creates an efficient environment for programmers to create smaller programs, which can be run by millions of computers.

System Software

System Software is a set of programs that control and manage the operations of computer hardware. It also helps application programs to execute correctly.

System Software are designed to control the operation and extend the processing functionalities of a computer system. System software makes the operation of a computer more fast, effective, and secure. Example: Operating system, programming language, Communication software, etc.

Application Software

Application Software is a program that does real work for the user. It is mostly created to perform a specific task for a user.

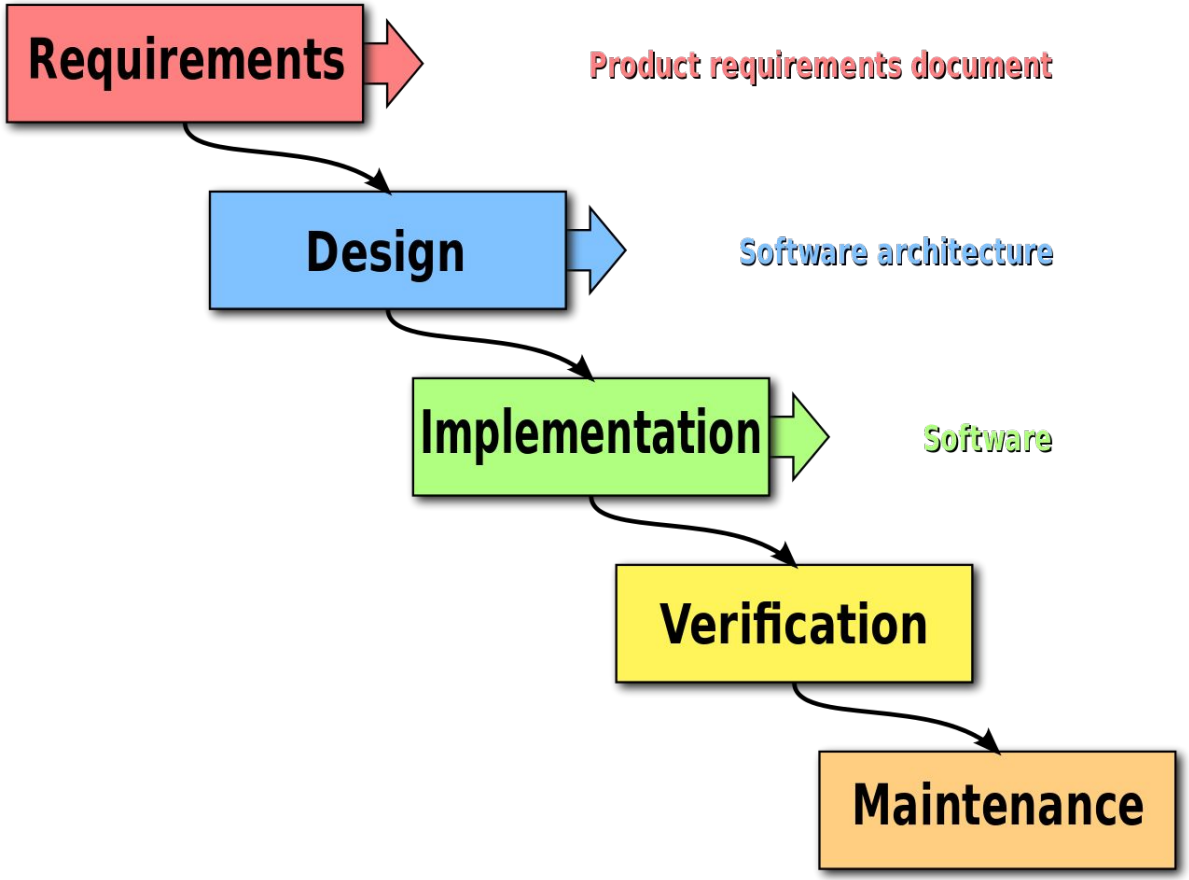
Application Software acts as a mediator between the end-user and System Software. It is also known as an application package. This type of software is written using a high-level language like C, Java, VB. Net, etc. It is a user-specific and is designed to meet the requirements of the user.

You can also install multiple Application Software on a single System Software. You can store this kind of software on CDs, DVDs, flash derive, or keychain storage devices. Example: Word-processing, Spreadsheet, Database, etc.

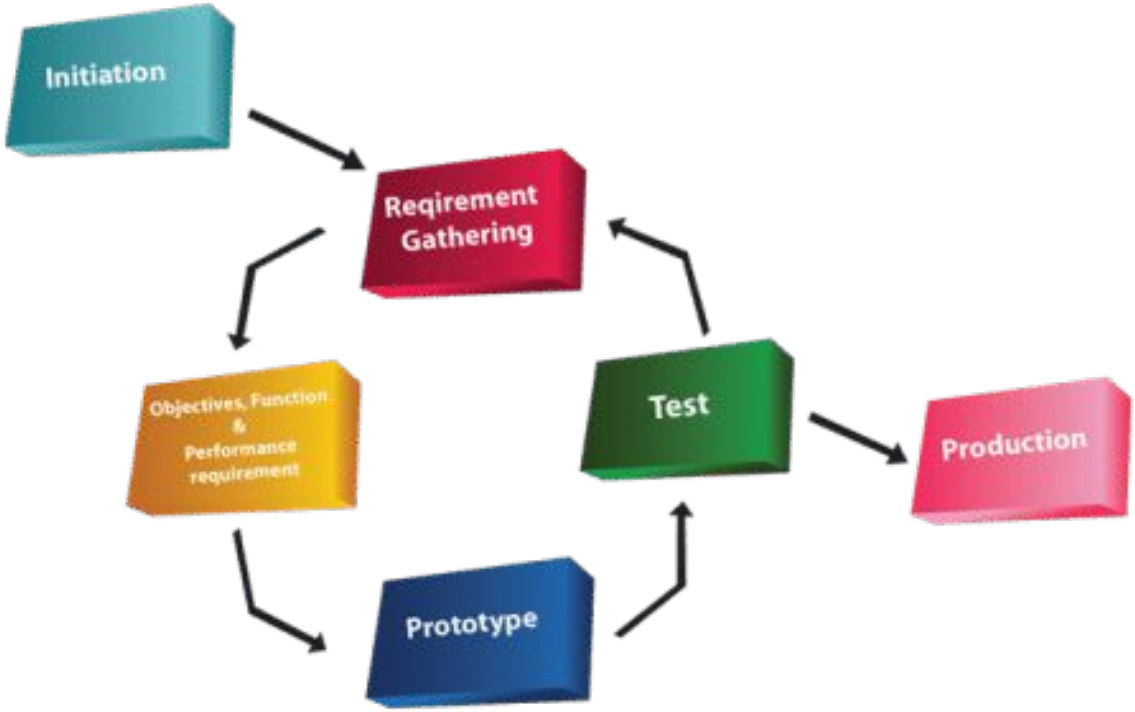
Software Development Model

- Waterfall Model
- Iterative / Prototype Model
- Spiral Model

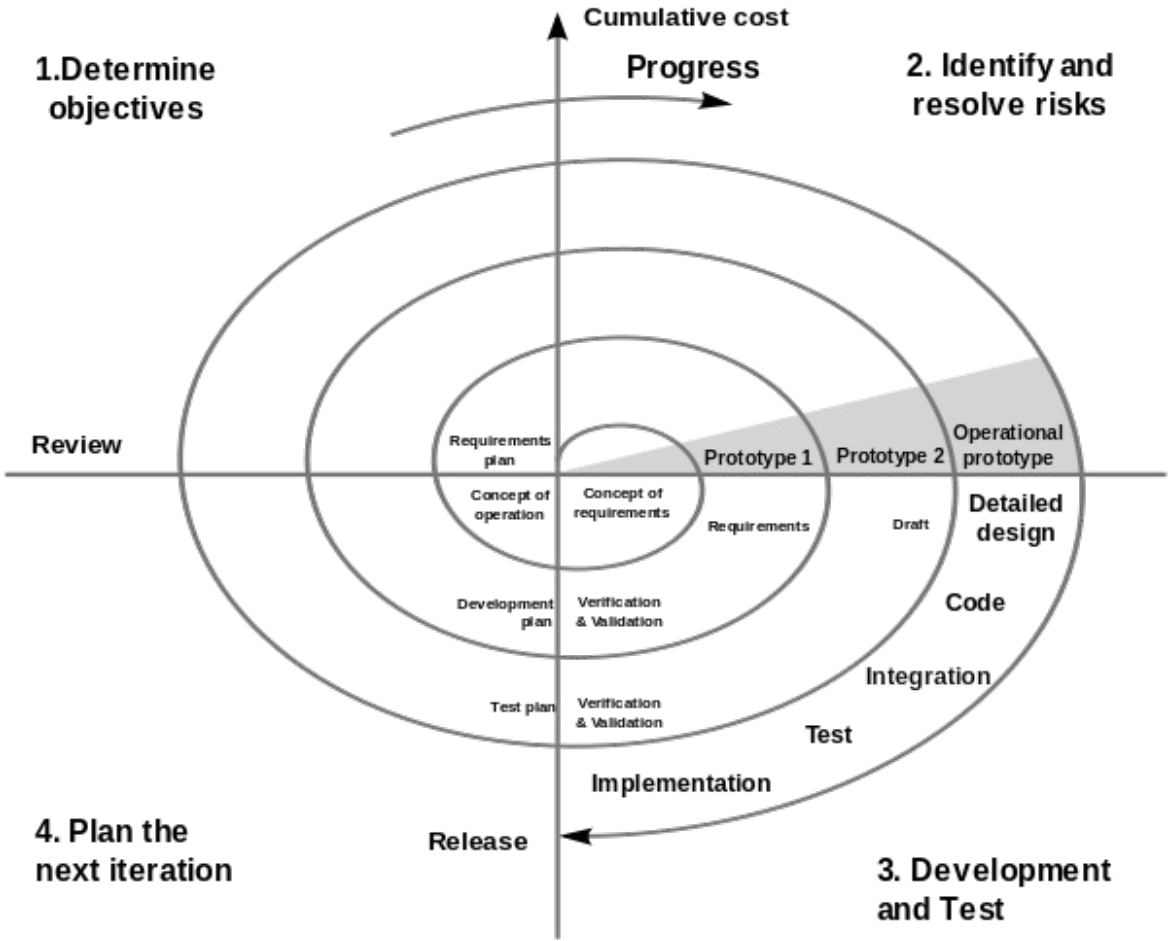
Waterfall Model



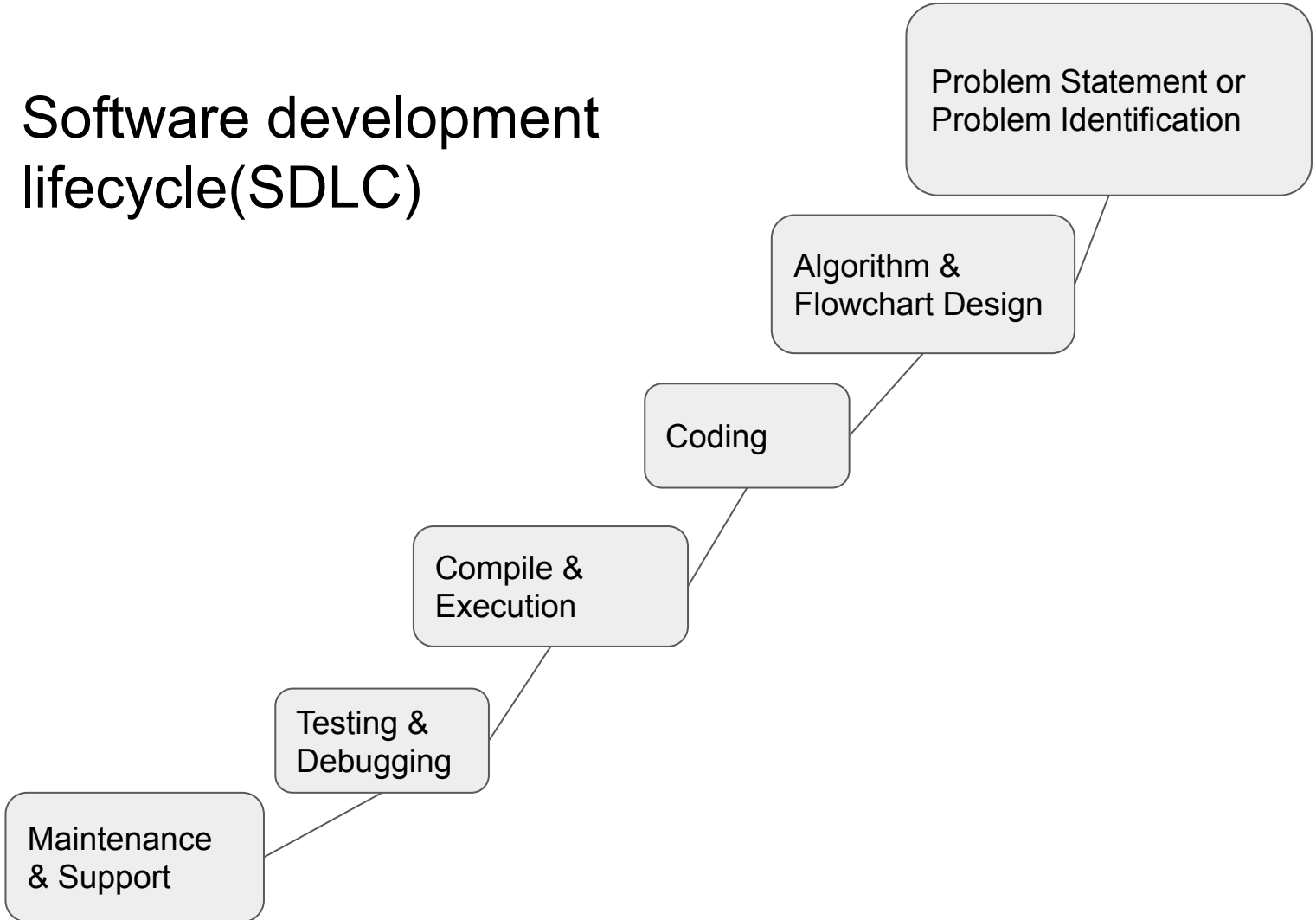
Prototype Model



Spiral Model



Software development lifecycle(SDLC)



Program

Basic commands that instructs the computer system to do anything are called instructions.

An organized list of instructions, that when executed, causes the computer to behave in a predetermined manner is a program. A program is like a recipe. It contains a list of ingredients (called variable) and a list of directions (called statements) that tells the computer what to do with the variables.

The variables can represent numeric data, text or graphical data. Without computer programs computer are useless.

Programming Language

- A set of rules that provides a way of instructing the computer to perform certain tasks is called programming language
- A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.
- A programming language is a formal language comprising a set of strings that produce various kinds of machine code output.

Types of Programming Language

- Low level or Machine level language
- Assembly level language
- High level language

Low Level language

- A low level language are mostly written with 1’s or 0’s directly understandable by CPU

Advantages

- It is the only language that computer understands and executes without the help of translator
- Very fast in execution

Disadvantages

- Difficult to write complex programs
- Can be prone to errors
- Less user friendly
- Takes a lot of time and difficult to modify

Assembly level language

- The assembly programming language is a low-level language which is developed by using mnemonics that converts assembly language to machine level language.

Advantages

- Comparatively easy to use than machine level language
- Less error prone than machine level language
- More control on hardware

Disadvantages

- Machine dependent
- Takes time to learn
- Slow development time
- Has no support for modern software engineering technology

High Level Language

- Programmer friendly language
- Needs compiler or interpreter to convert high level code to machine level code

Advantages

- Easy to understand
- Easy to debug
- Can run on any platform
- Is less memory efficient

Disadvantages

- Not machine friendly language

Language Processor

A language processor is a special type of program or software that converts or translates program codes or source codes into machine codes i.e. in the form of 1’s or 0’s.

- Compiler
- Assembler
- Interpreter

Compiler

- It is a program that converts entire source code or program codes into machine level codes at once
- It generates error only after scanning the whole code
- Eg: C, C++, Java etc uses compiler to convert high level code to machine level code

Assembler

- Assembler is used to convert assembly level code to machine level code
- The source program contains input of assembly level instructions
- It uses opcode for the instructions. An opcode basically gives information about the particular instruction. The symbolic representation of the opcode (machine level instruction) is called mnemonics.

Interpreter

- It takes high level code and translates into machine level code line by line.
- Takes less time to analyze the line code but the total time to analyze a whole program increases
- Debugging is easier as the program will be continued only after the errors are dealt with.
- Example: Python, Perl etc uses interpreter to convert program code into machine code

Program Errors

- It is something in the program code that prevents a program from running and compiling successfully.
- It is also known as bugs
- Logical Error
- Syntax Error
- Run time Error

Logical Error

- It occurs due to some error in logic while writing a program that leads to undesired or wrong outputs
- But it does not terminate the program abnormally
- An example of a logical error would be dividing by 2.54 instead of multiplying to convert inches to centimeters.

Syntax Error

- A syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a particular programming language
- It is an error related to the grammar or specific syntax in which a code is to be written.
- Examples of syntax errors are missing a comma or a quotation mark, or misspelling a word

Run time error

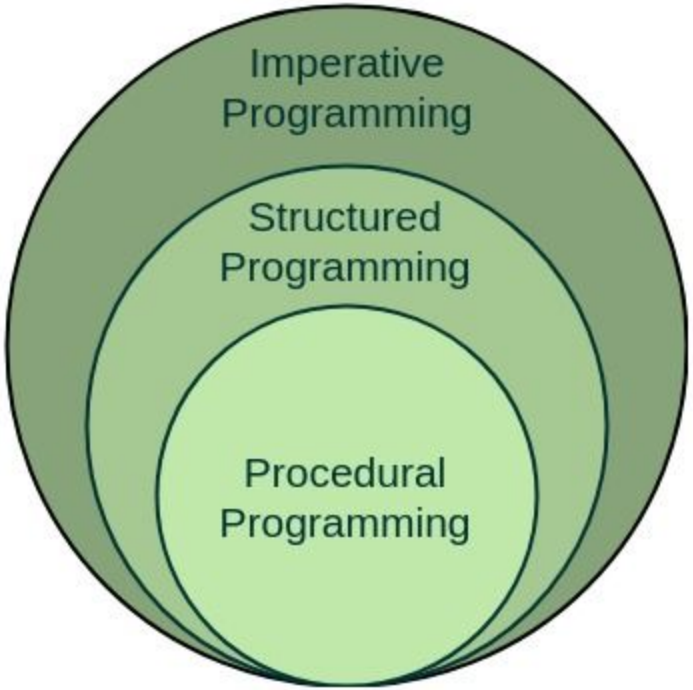
- Runtime error refers to an error that takes place while executing a program
- These errors occur due to illegal operation performed in the program

Common types of runtime error

- Memory leak
- Undefined object error
- Input Output Error
- Division by 0 error

Structured Programming

- It can be defined as a programming approach in which the program is made as a single structure
- It means that the code will execute the instruction by instruction one after the other
- The instructions in this approach will be executed in a serial and structured manner

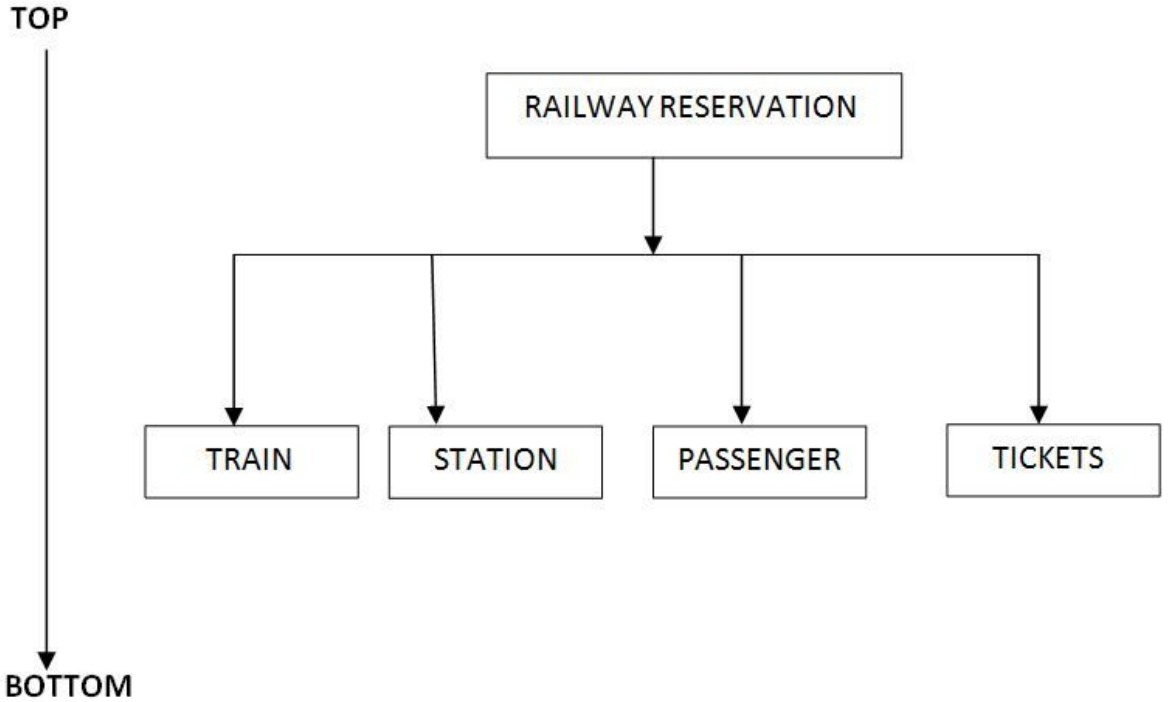


Advantages of Structured Programming

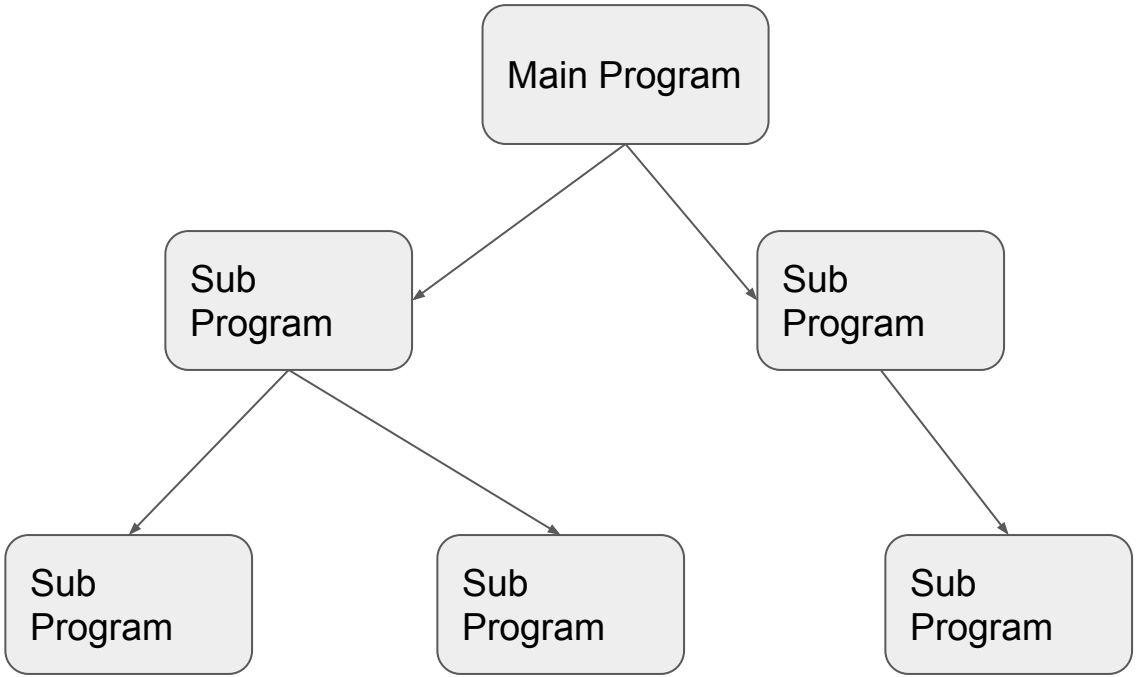
- Easier to read and understand
- User Friendly
- Easier to Maintain
- Mainly problem based instead of being machine based
- Development is easier as it requires less effort and time
- Easier to Debug
- Machine-Independent, mostly.

Top Down approach

- In this approach we focus on breaking the big program into smaller program
- If the sub program is difficult, we further break it into smaller program
- This is mainly used in C, Fortran etc.
- Example:

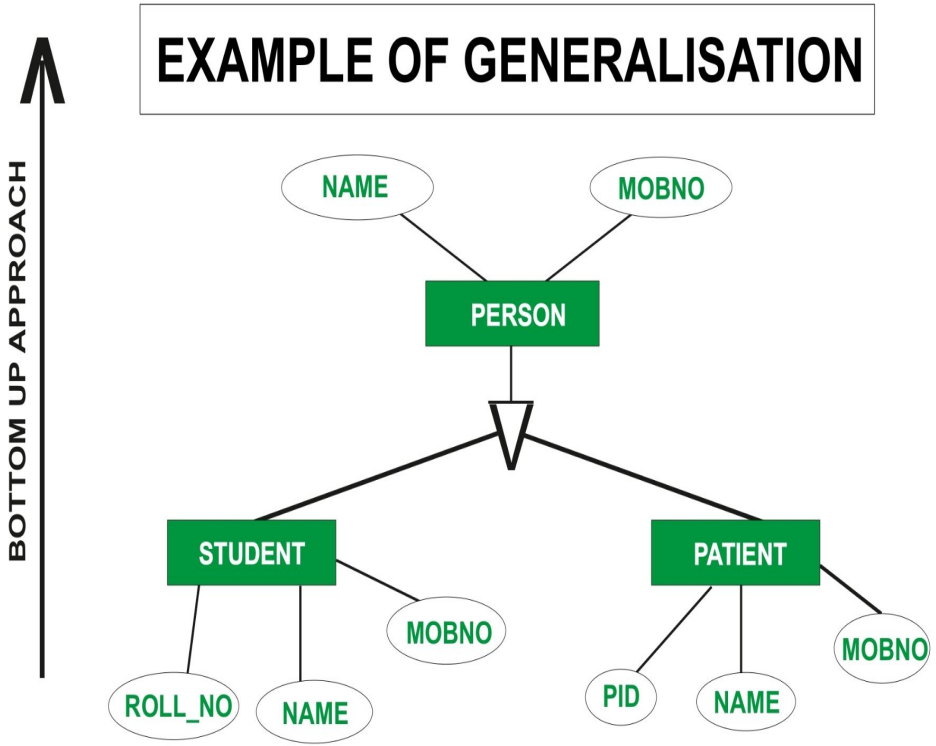


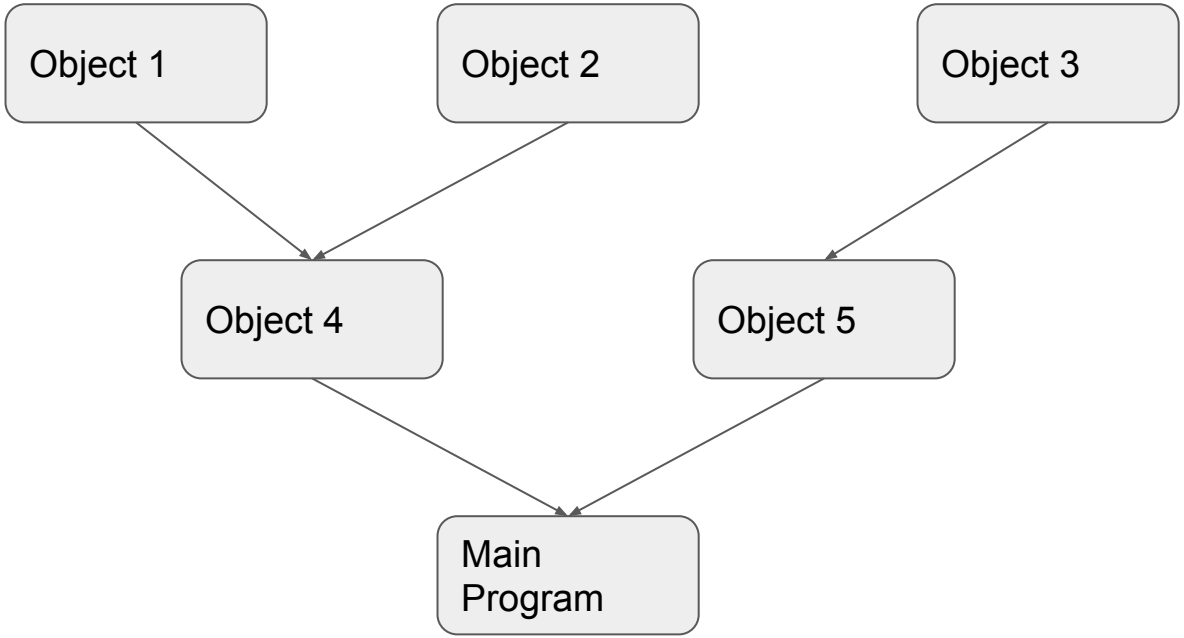
Top Down Approach



Bottom Up Approach

- In bottom up approach, we will create small problems first, then will solve the smaller problems.
- Then we will try to integrate it into a whole program and complete the solution
- Mainly used by object oriented programming languages like C++, Python, Java etc.
- Example





Bottom Up Approach

Problem Solving

Problem is defined as the difference between an existing situation and a desired function, that is, in accordance with calculation

Solution is a desired situation and has the simplest form. If a problem is solved by computing using machine called computer, then such process is called Problem Solving.

First and foremost step of solving a problem is knowing about the nature of problem in order to solve it. Almost all types of problems can be solved with computer.

However, correct formulation of the problem in computer understandable terms is essential to formulate the problem and solve it using computer.

Understanding of problems, Feasibility and Requirement Analysis

Problems can be of various nature and we need to understand about the problems first to be able to solve it using computer.

Once the problem has been understood and we can solve it using computer the feasibility of the problem solution needs to be determined.

We can solve a problem using various logic but what needs to be worked out is the most feasible solution that can be derived with the least amount of complexities.

Another major portion to solving problems using computer is the analysis of the requirements.

The solution to the problem can be solved however the requirements required is beyond the possible technologies of today.

Then the solution may not be feasible to the problem.

Algorithm

An algorithm is defined as a set or ordered steps or procedures necessary to solve a problem.

It is a step wise presentation of procedures of program in simple English

To be an algorithm, the set of steps must be unambiguous and have a clear stopping point. Each step tells what task is to be performed

Algorithm for addition of two numbers

1. Start
2. Read two numbers a and b
3. Perform addition, sum = a+b
4. Display sum
5. Stop

Pseudocode

English like steps that describe the solution.

Pseudocode is an artificial and informal language that helps programmers develop algorithms.

Pseudocode is a ”text-based” detail (algorithmic) design tool.

The rules of Pseudocode are reasonably straightforward. All statements showing ”dependency” are to be indented.

Flowchart

Pictures Detailing with specific blocks detailing out the logical flow of the solution.

For a better understanding of an algorithm, it is represented pictorially.

The pictorial representation of an algorithm is called a Flow Chart.

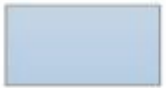
Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Figure 2: Flow chart Basic Symbols

Program coding

Once algorithm and flowchart has been developed the task now remains is to write programs using some high level programming language.

C, C++, Java, Python are the most popular programming language to develop programs.

PHP and ASP remain a popular choice for developing web based applications.

Testing and Debugging

Once the program code has been written in a selected programming language of choice the next task to complete is the testing and debugging.

Testing and debugging helps to find the problems associated with the program behavior under normal and abnormal circumstances.

Extensive testing like white box testing and black box testing, integrationtesting needs to done before the program is deployed into the real world scenario.

Implementation

After a program has been written and tested it needs to be implemented to the target environment to solve the problem.

Various needs of physical hardware and accessories required by the program to solve the intended problem needs to be present upon implementation.

Once the program is implemented it starts to work.

Evaluation and Maintenance

The evaluation of the performance of the program needs to be done at frequent interval once the program or software is implemented.

The advent of new technology, upscaling and downscaling of the business, request for the change from customers, finding of a new bug are the major reasons for the maintenance and change of the softwares.

Once changes have been continuous monitoring of the software performance needs to done to discover the flaws in the software.

Documentation

The same workforce that developed the program may have left the project and gone in search of opportunities.

So a program must be well documented in order for the new people to understand how the software was developed and how can it be modified.

The documentation start from the very beginning of the problem formulation to the very end of the Evaluation and Maintenance.