Date / /
Page

Unit - 1

Introduction to Computers

A computer is an electronic device that manipulates data or information according to set of instructions is called program.
A computer is a device capable of performing computations and making logical decisions at speed millions and even billion of times faster then human beings can.

Defination of Computer :-

Generally computer is the combination of hardware and software which convert data into information.

Hardware → Hardware refers to the physical and visible components of the system such as a monitor, CPU, keyboard and mouse.

Software → It refers to a set of instructions which enable the hardware to perform a specific set of tasks. OR It is a collection of data.

Data :- Data is information that can be interpreted and used by computers. It is a collection of facts, such as numbers, figure and text on a piece of paper are known as Data.

Processing - It is the sequence of action taken on data to convert it in to information.

→ Information — When data is processed, organised and structured, presented in a given context so as to be useful then is called information.

→ Instruction → It is a command given to a computer in the computer language by the users.

→ Program :— It is a set of instructions given to a computer in order to perform some task. OR Application are program.

→ Characteristics of the Computer :—

(1) Speed → Generally a computer can carry out 3.4 million instructions per second.

(2) Accuracy → Computers exibert a very high degree of Accuracy error that may be error are usually due to incorrect data, wrong instruction or bugb in cheat all human errors.

(3) Reliability → Computer can carry out same types work rapidly without throughing up error due to tiredness.

(4) Storage Capacity → Computer can store a very large amount of data at a fraction of a cost of traditional storage of file.

Date:

Page:

Advantage of Using Computer :-

→ Computers can do the same task repetitively with same accuracy.

→ Computer can take a routine task wide using human resources more better function.

→ Computer do not get tired or broken.

Disadvantage of Using Computer :-

Computer have no intelligence they follow instruction blindly considering the out come.

Regularly electric supply is necessary to make computers work which complete proof difficult every where specially in developing nation.

Booting :- Starting a computer or a computer embedded device is called booting.

Booting take place in two steps.

The first program a set of instruction that runs when computer is switched on spot BIOS (Basic input output system) BIOS is the form a where.

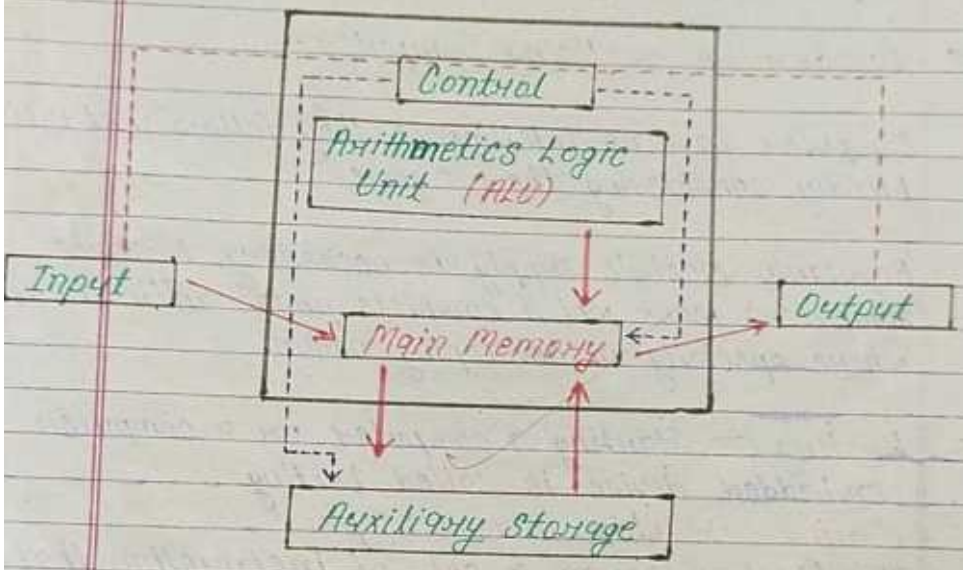
There are two types of booting :-

(1) Old Booting → When the system is started by the switching on the power supply. are called old Booting.

Date: / /
Page:

Warm Booting — When system is already running and need to be restart or rebooting is called warm booting.
Warm bootingisfaster than old booting because BIOS is not reloaded.

Block Diagram of Computer



In the above diagram both control unit (CU) and Arithmetics Logic Unit (ALU) combinedly called as central processing unit (CPU).

Let discuss about all the parts as included in above diagram one by one.

(1) CPU (Central processing unit) → It is the brain of the computer system all the major

calculation and comprising one made in side the (CPU) and it is also responsible for activation and controling the operation of other unit.

→ This unit consists of two major components that is Arithmetics logic unit (ALU) and Control pr unit (CU).

⇒ (ALU) Arithmetics Logic unit :→

There Arithmetics Logic unit performance all the arithmetic operations such as addition, subtraction, multiplication and division.

⇒ Control Unit (CU) → Control unit of the CPU controls the entired operations of the computer. It also controls all devices such as memory, input and output (I/O) devices connected to the CPU. Control unit instructions from memory, decodes the instruction interpretation. To know what the task the task one to be performed and saying sayings suitable controls signal to the other components to perform for the necessary steps to executes the instruction.

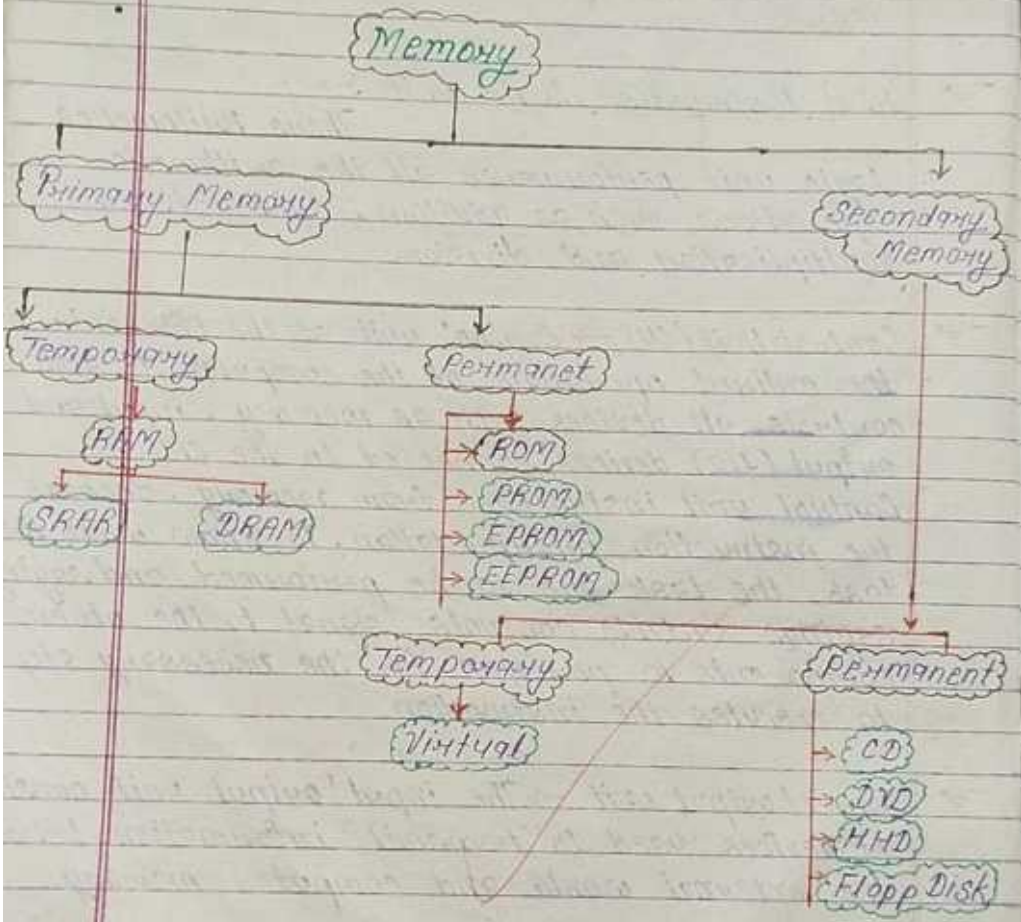
⇒ Input/output unit → The input/output unit consist of devices used to transmit information between the external world and computer memory.

→ Memory unit → Memory unit is on inital components of a digital computer it is where

Date: __/__/____
Page: ____

all data intermediate and find result are storage is called memory unit.

The memory unit is used to hold the instruction to be executes and data to be process.



→ Disk Storage Unit – Data and instructions enters in to computer system through inputs device have to store in side the computer before actual processing start.

→ Two types of storage unit -
Primary storage unit
Secondary storage unit

→ Primary Storage unit – Primary memory has directly with inputs units and output units it stores the input data calculation results.

→ Secondary Memory – The primary storage is not able to store data permanently for future use so some other types of storage technology is required to stored the data permanently for long time it is called secondary or auxilliary storage.

→ Primary Memory :-

RAM – RAM is stand for Random Access Memory It is the internal memory that can be read from as well as written to this memory is obtain associated with volatile types of memory. is can hold data only on temporary basis because it requires a continous flow of electrical current.

To main form of RAM are as follow –

Date: __/__/____
Page: _____

SRAM (Static Random Access memory) *
DRAM (Dynamic Random access memory)

3. ROM - In ROM (Read only Memory) information once stored remain fixed it can not be changes. *
So, ROM can only be read and used, the context of ROM remain stored even this power is turned off.

There are 3 Types of ROM 3

→ PROM - Programmable Read Only Memory it is a computer memory cheap cable of being programmed after which has been created but once of the PROM has been programmed the information. #

→ EPROM - EPROM (erasable programmable readonly memory) is memory that does not lose its data when the power supply is cut off. The data can be erased and the chip reprogrammed by shining an intense ultraviolet (UV) light through a window designed into the memory chip. #

#

EEPROM - EEPROM is a type of non-volatile ROM that enables individual bytes of data to be erased and reprogrammed.

Date: / /

Page:

* CD — The full form of CD is the compact disc. It is a flat, tiny-rounded storage unit that can store up to 700 MB of information and is 4.75 inches in diameter.

* DVD — The full form of DVD is Digital Versatile Disc. DVD is a digital optical disc storage and it is used to store high-capacity files, such as top standard videos and movies.

⇒ Types of Computer :-

Personal Computer — A personal computer can be defined as a small, relatively inexpensive computer designed for an individual user. Personal computers are based on the microprocessor technology that enables manufacturers to put an entire CPU on the chip. At home, the most popular use for personal computers is playing games and surfing the Internet.

Minicomputer — It is a midsize multi-processing system capable of supporting up to 250 users simultaneously.

Mainframe Computer — Mainframe is very large in size and is an expensive computer capable of supporting hundreds or even thousands of users simultaneously. Mainframe executes many programs concurrently.

Date: / /

Page:

Super Computer – Super computers are one of the fastest computers currently available. Supercomputers are very expensive and are employed for specialized applications that require immense amount of mathematical calculations.

For example – Nuclear energy research, electronic design, and analysis of geological data etc. Param 8000, 12000.

Unit-2

Basic Computer Organization

Generation of Computer – In the computer language generation is a set of technology it provides for the growth of the computer technology. There are totally five generation of computer we today discussed as following :-

(1) First generation –

Duration – 1942 - 1955

Technology - Vaccum tube

→ Used as a calculation in minisecond.
→ Bulky in size and complex design.
→ Required large room to place it.

→ A computer using vaccum tube is very slow in execu-ting programmes compared to present day computer.

(2) Second Generation —
Duration — 1955 - 1964
Technology - Transistor
→ Ten times smaller in size then first generation systems.
→ Less heat then first generation computer.
→ Consumed less power than first generation system.
→ Computer whose done calculation in micro second.
→ Air condition is also required.

(3) Third generation —
Duration - 1965 - 1975
Technology - IC chip (Integrated circuit)
→ Smaller in size then first and second generation computer.
→ Perform more fast calculation then second generation
→ It is general purpose computer.

(4) Fourth Generation —
Duration - 1975 - 1985
Technology - Micro processor chip
→ Based on VLSI technology used.
→ Small in size.
→ Much faster than previous generation
→ Minimum handward maintains is required.
→ No requirement of air condition.

(5) Fifth Generation -

Duration - 1989 - Present

Technology - ULSI

Micro processor chip.

VLSI - Very large scale integration

→ Much smaller and handly

→ Based on ULSI cheap which contain 100 millions electronic components.

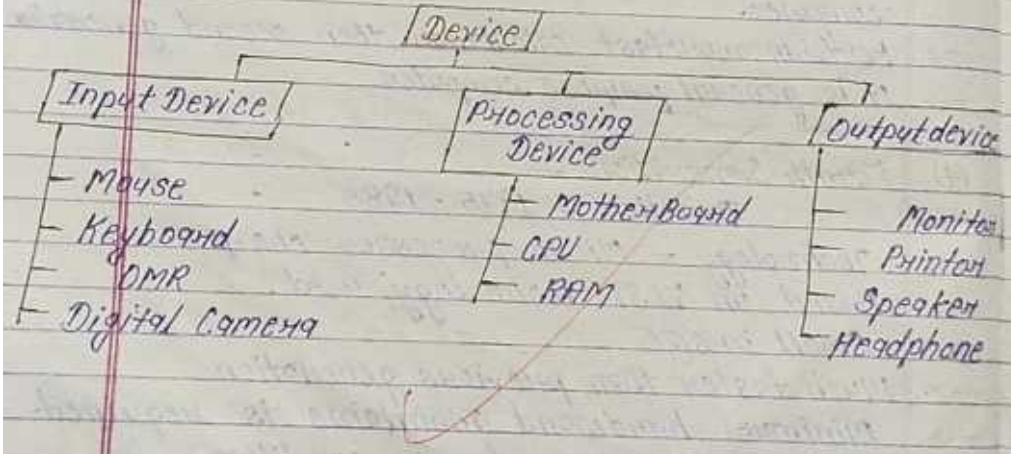
→ The speed of operations increased.

→ Air condition is not required.

→ The more user friendly interface and with the multimedia features.

Example -

3 Device - Device is an electronic machine that is used to complex task in computer machine.



Date: __/__/____
Page: _____

3. Input Device — The device which are used to interg data in the computer system is known as input device.

Example - Mouse, Keyboard, OMR, Digital computer.

Keyboard — There are total 104 and in this 22 special keys on keyboard such as following:

1. ESC (Escape key)
2. F1 to F12 (Function Key)
3. Tab
4. Caps Lock
5. Shift
6. Ctrl (Control)
7. Window
8. Alternate
9. Space Bar
10. Enter
11. Backspace
12. Scroll Lock
13. End
14. Insert Lock
15. Delete
16. Page Down
17. Page Up
18. Menu Key
19. Arrow Key
20. Page Break
21. Number Lock

Date / /
Page

3. Output device → Output device is a machine that is appeared a result after the processing. There are many types of output device in a systems.
Example – Monitor, Printer, Speaker, projector, head phone etc.

* Components of Computers –
There are basically two types of computer components :–

(1) Hardware → Hardware is a physical part of the computer system it can be touch & seen. Hardware cannot perform any task without software.

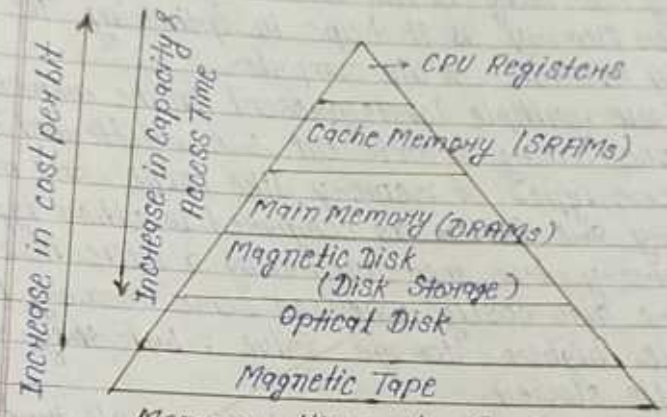
(2) Software – Software is collection of program that is perform some specific task that provide the instruction for telling a computer what to do and how to do it.

(a) System Software → System software are the directly responsible for controling integrated and managing the individual hardware components of a computer system.

(b) Operating System – An operating system is system software that manages computer hardware and software resources, and provides common services for computer programs.

Memory Hierarchy

In the Computer System Design, Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time.
The Memory Hierarchy was developed based on a program behavior known as locality of references.
The figure below clearly demonstrates the different levels of the memory hierarchy.



Memory Hierarchy Design

→ Application Software — It is a computer software design to help the users to perform singular or multiple task. It is set of instruction or program design for specific uses are application the enable the user to interact with a computer.
Application software are also called the end user program.

Date: / /
Page:

System utilities - These programs perform task related to the maintains of the computer system these are the package which are loaded into computer during time of installation of operating system.

Example - Back up utilities, disk cleaners, antiviruses, disk fragmenter.

→ Why Memory Hierarchy is Required in the system:

Memory Hierarchy is one most required things in Computer Memory as it helps in optimizing the memory available in the computer.
There are multiple levels present in the memory, each one having a different size, different cost etc. some types of memory like cache, and main memory are faster as compared to other types of memory but they are having a little less size and are also costly whereas some memory has a little higher storage value, but they are a little slower.
Accessing of data is not similar in all types of memory, some have faster whereas some have slower access.

3 Types of Memory Hierarchy

This memory hierarchy design is divided into 2 main types:-

Date
Page

(1) External Memory or Secondary Memory –

Comprising of Magnetic Disk, Optical Disk, and Magnetic Tape that is peripheral storage devices which are accessible by the processor via an I/O Module.

(2) Internal Memory or Primary Memory –

Comprising of main Memory, Cache Memory & CPU registers. This is directly accessible by the processor

3 Characteristics of Memory Hierarchy

1. Capacity – It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.

2. Access Time – It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increase.

3. Performance – Earlier when the computer system was designed without a memory hierarchy design, the speed gap increased between

Date: / /
Page:

the CPU registers and Main memory due to large difference in access time. This results in lower performance of the system and thus, enhancement was required.

* Cost Per Bit — As we move from bottom to top in the Hierarchy, the cost per bit increases that is Internal Memory is costlier than External Memory.

⇒ Advantages of Memory Hierarchy :-

→ It helps in removing some destruction, and managing the memory in a better way.
→ It helps in spreading the data all over the computer system.
→ It saves the consumer's price and time.

Introduction to problem solving Concept —

Problem solving is a critical skill in computer science and programming. It refers to the process of finding solutions to problems or challenges by applying logic and critical thinking.

Some key concepts in problem solving -

* Understanding the problem — This involves carefully reading and comprehending the problem

statement and defining the problem in your own words.

2. Analyzing the problem – This involves breaking down the problems into smaller, more manageable parts and identifying the information and data required to solve the problem.

3. Formulating a plan – This involves coding the solution creating a step by step plan for solving the problem, including the methods and algorithms that will be used.

4. Implementing the plan → This involves coding the solution and testing it to ensure that it works as expected.

5. Evaluating the solution – This involves analyzing the solution to ensure that it's correct, efficient and meets the requirements of the problem.

Types of errors –

* Syntax errors – These errors occur when code violates the rules of the programming language. For example – using "print" instead of "printf" to display the data would result in a syntax error because the compilers or interpreters doesn't recognize 'print". It is usually detect syntax errors and prevent the code from executing until the error is fixed.

Date: / /
Page:

* Logical errors — These errors are a type of runtime error that occur when there are miscalculations in the logic code. This can happen when the wrong formula is used or the wrong variables are called. Logical errors can result in incorrect outputs from the program.

* Runtime errors — These errors can be caused by external factors, such as user input, network connection, or hardware failure. They can also be caused by failure. They can also be caused by internal factors, such as memory leaks, buffer overflows, or infinite loops.

* Semantic errors — These errors can be dangerous because they might not be noticed by programmers or users.
Programs can go for years with undetected semantic errors, and no one will realize that the program is producing incorrect results.

* Compilation errors - These errors are usually caused by semantic or syntactical errors.
You can test your code against custom input to ensure that it doesn't have any of these errors.

* Interface errors - These errors indicate a disconnect between how a program is intended to be used and how it's actually used by users, other software components or system

Date:
Page:

Unit-3 Techniques of problem solving

→ What is a Flowchat and its Types?

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. A flowchart is a pictorial way of representing steps that are preferred by most beginner level programmers to understand algorithms of computer.

→ A flowchart is a picture of boxes that indicates the process flow sequentially. To draw a flowchart certain rules need to be followed which are followed by all professionals to draw a flowchart and are widely accepted all over the countries.

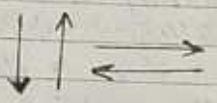
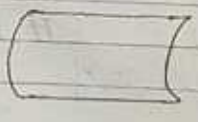
→ Flow chart Symbols

These are six basic symbols commonly used in a flowchart.

NAME	SYMBOL	THEIR FUNCTIONS
Terminal/ Terminator		This symbol indicates the beginning and ending of a program in a flowchart.
Data or Input/output		It is used to indicate input and output of a program. It takes input and show output.

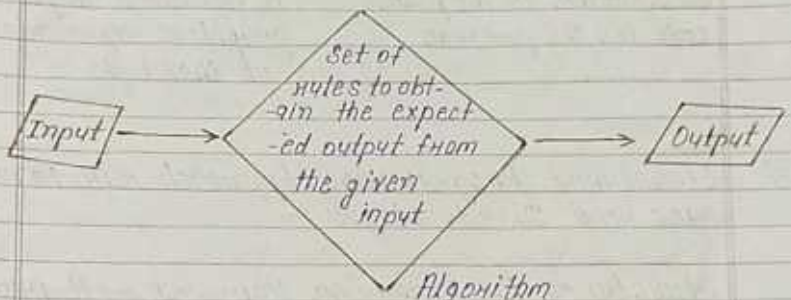
Date: / /

Page:

Process		This symbol indicates a process, action or any type of operation, like addition, multiplication, division and subtraction.
Decision		It is used to raise a question or a condition in a program that can have Yes/No as an answer. It is known as condition box.
Flow line/ Arrow		Flow arrow show the direction of flow in a program. These lines are also used to connect the various symbols in a flowchart.
Connectors		These are used to connect breaks in the flowchart.
Document		
Predefined process		
Stored Data		
Comment or Annotation		

Date: / /
Page:

→ Algorithm — The word Algorithm means " a process or set of rules to be followed in calculations or other problem-solving operations ". Therefore algorithm refers to a set of rules / instructions that step by step define how a work is to be executed in order to get the expected results. Example →



Algorithm

→ Difference between algorithm and flow chart:

Algorithm	Flowchart
An algorithm is a step-by-step procedure to solve a problem.	A flowchart is a diagram created with different shapes to show the flow of data.
The algorithm is complex to understand.	A flowchart is easy to understand.
In the algorithm, plain text is used.	In the flowchart chart, symbols / shapes are used.
The algorithm is easy to debug.	A flow chart is hard to debug.

The algorithm is difficult to construct	A flowchart is simple to construct.
The algorithm does not follow any rules.	The flowchart follows rules to be constructed
The algorithm is the pseudo code for the program	A flowchart is just a graphical representation of that logic.

3 Structured Programming Approach with Advantages and Disadvantages :-

Structured Programming Approach — 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. It doesn't support the possibility of jumping from one instruction to some other with the help of any statement like GOTO etc. The instructions in this approach will be executed in a serial and structured manner. The languages that support Structured programming approach are:

- C
- C++
- Java
- C#

Date: / /
Page:

Imperative Programming
Structured Programming
Procedural Programming

The structured program mainly consists of three types of elements:

→ Selection statements
→ Sequence statements
→ Iteration statements.

∃ Advantages of Structured Programming Approach

→ 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.

∃ Disadvantages of Structured Programming Approach

→ It is Machine-Independent, so it takes time to convert into machine code

→ The converted machine code is not the same as for assembly language.

→ The program depends upon changeable factors like data types. Therefore it needs to be update with the need on the go.

Programming Methodology

When programs are developed to solve real-life problems like inventory management, payroll processing, student admissions processing etc they tend to be huge and complex. The approach to planning for software development and controlling the development process is called programming methodology

Types of Programming Methodologies

There are many types of programming methodologies. But two most common types of programming methodology.

(1) Procedural Programming - Problem is broken down into procedures, or blocks of code that perform one task each. All procedures taken together from the whole program. It is suitable only for small programs that have low level of complexity.
Example - For a calculator program that does addition, subtraction, multiplication, division, square root and comparison, each of these

operations can be developed as separate procedures. This methodology is suitable for solving small problems.

⇒ Object-oriented Programming – Here the solution revolves around entities or objects that are part of problem. The solution deals with how to store data related to the entities, how the entities behave and how they interact with each other.
Example- If we have to develop a payroll management systems, we will have entities like employees, salary structure, leave rules etc around which the solution must be built.

⇒ Top-down approach and Bottom-up Approach

(i) Top-down or Modular Approach – In this approach the problem is broken into smaller units, which may be further broken down into even smaller units. Each unit is called module. Each module is a self-sufficient unit that has everything necessary to perform its task. The following

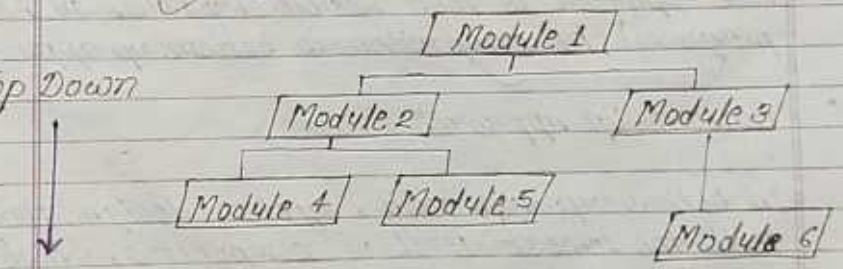


illustration shows an example of how you can follow modular approach to divide the

problem into different module while creating a program.

→ Advantages of top-down approach –

→ Using this approach, the problem solving process of big problem become easy.
→ This approach utilize the resources in a proper manner according to the project.
→ Testing and debugging is easier and efficient.
→ In this approach, project implementation is smoother and shorter.
→ This approach is good for detecting and correcting time delays.

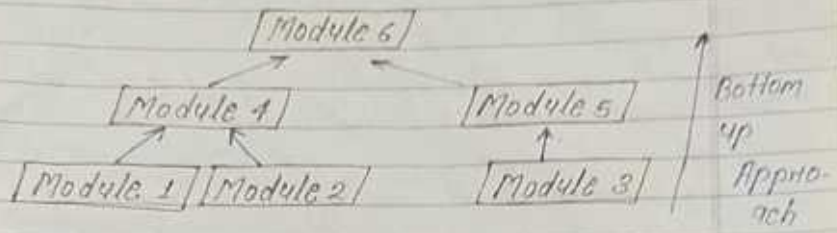
→ Disadvantages of Top-Down approach

→ This approach is not suitable for solving highly complex problems.
→ This approach is less flexible (updation in the program) in comparison to bottom up approach.

→ Bottom-Up Approach

In bottom up approach, system design starts with the lowest level of components, which are then interconnected to get higher level components. This process continues till a hierarchy of all system components is

However, in real life scenario it is very difficult to know all lowest level components at the beginning.



Advantages of Bottom Up Approach :-

Flexible → This approach is very flexible addition of new features and modification of features is easiler.

Realiability → This approach is more realiable because in this case each part of the program is independently created and tested tested first then it is integrated.

Lower cost → In this approach the component are reused therefore the cost of the software development decrease.

Disadvantages of Bottom Up Approach :-

* It is difficult to evaluated the progress of the project in this approach.

Date: __/__/____

Page: _____

* The idea about the final product is obtain at end only.

Unit-4

Date: / /

Page:

Overview of Programming

A programming language is a formal computer language or constructed language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behavior of a machine or to express algorithms.

Introduction of Python :-

* Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language.
* Python is easy to learn yet powerful and versatile scripting language which makes it attractive for application development.
* Python is supports multiple programming pattern, including object oriented programming, imperative and functional programming or procedural styles.

History of Python

Python was conceptualized by Guido Van Rossum in the late 1980s. Rossum published the first version of Python code (0.9.0) in Feb. 1991 at the CWI (Centrum Wiskunde & Informatica) in the Netherlands, Amsterdam.

Python is derived from ABC programming language, which is a general purpose

programming language that had been developed at the CWI.

Difference between Compiler & Interpreter

Compiler	Interpreter
(1) Compiler works on the complete program at once. It takes the entire program as input.	(1) Interpreter program works line by line. It takes one statement at a time as input.
(2) Compiled programs take more memory because the entire object code has reside in memory.	(2) Interpreter does not generate intermediate object code or machine code.
(3) Compile once and run any time. compiled program does not need to be compiled every time.	(3) Interpreter does not generate intermediate object code. As a result. interpreted programs are more memory efficient.
(4) A compiled language is more difficult to debug.	(4) Interpreted program are interpreted line by line every time they are run. Debugging is easy.

PYTHON VERSIONS

Python 1.0
Python 2.0
Python 3.0

Date / /
Page

* APPLICATIONS OF PYTHON →

Machine Learning
GUI Applications
Web frameworks like (used by Youtube, Instagram
Image processing (like OpenCV, Pillow)
Test frameworks
Multimedia
Scientific computing
Test processing

Types of errors in python

1. Syntax errors → Syntax errors are due to the fact that the syntax of the Java language is not respected.

Example - Missing semicolon:
int a = 5// semicolon is missing
Compiler message:

Java:20:' ;' expected
int a = 5

2. Semantic errors -
Semantic errors due to an improper use of program statements.

Example → Use of a non initialized variable:
int i;
i++; // the variable i is not initialized

Date: / /
Page:

② Errors in expressions:
String s = "...";
int a = 5 - s; // the - operator does not support arguments of type String.

③ Logical errors — Logical errors are caused by the fact that the software specification is not respected. The program is compiled and executed without errors, but does not generate the requested result.

Example - Errors in the performed computation
public static int sum (int a, int b){
return a-b;
}
// this method returns the wrong value wrt the specification that requires
// to sum two integers.

From the point of view of when errors are detected we distinguish:

1. Compile time errors: syntax errors and static semantic errors indicated by the compiler.

② Runtime errors: dynamic semantic errors, and logical errors, that cannot be detected by the compiler (debugging).

Date: / /

Page:

Structure of a Python Program

Python Statements –

The interpreter reads and executes the statements line by line that is sequentially. Though there are some statements that can alter this behaviour like conditional statements.

Mostly, python statements are written in such a format that one statement is only written in a single line.

The interpreter considers the 'new line character' as the terminator of one instruction.

Instructions that a Python interpreter can execute are called statements.

Example –

a = 1 is an assignment statement

if statement

for statement

while statement etc

Multi-line statement –

In Python end of a statement is marked by a newline character. But we can make a statement extend over multiple lines with the line continuation character (\)

```
a = 1 + 2 + 3 + \
    4 + 5 + 6 + \
    7 + 8 + 9
```

Date: / /
Page:

Python Indentation -

Most of the programming languages like C, C++, Java use braces { } to define a block of code.

Python uses indentation.
A code block (body of a function, loop etc) starts with indentation and ends with the first

Python Comments - what is going on side program
In Python, we use the hash (#) symbol to start writing a comment.

Python Variables -
A variable is a named location used to store data in the memory. It is helpful to think of variable as a container that holds data which can be changed later throughout programming.

Example -> N1 = 10
N2 = 10.5

* Constant - Is a type of variable which can not be changed. It is containers which hold data or that information that can not be changed.
PI = 3.14
Gravity = 9.8

Date: / /

Page:

Literals- Literals is a raw data given in a variables or constant.

In Python there are many literals

Example-

```
a = 0b1010 # Binary literals
b = 100 # Decimal
c = 0o310 # Octal
d = 0x12c # Hexadecimal
# Complex - x = 3.14j
# float - float - 1 = 10.5
        "   - 2 = 1.5e2
```