KTU NOTES

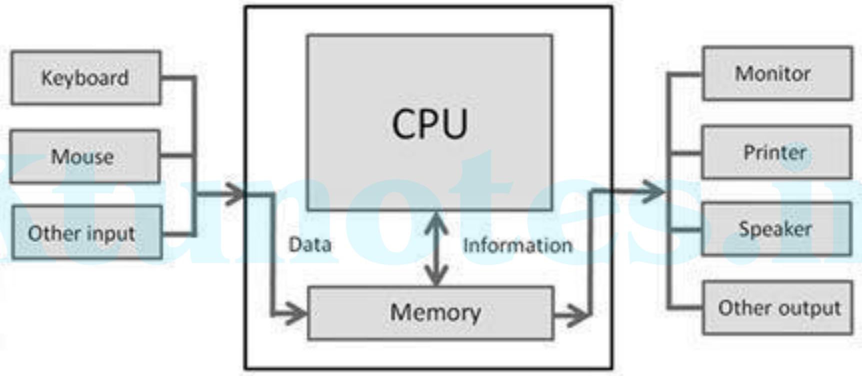
Module I

Basics of Computer Hardware and Software, Basics of Computer Architecture: processor, Memory, Input& Output devices. Application Software & System software: Compilers, interpreters, High level and low level languages Introduction to structured approach to programming, Flow chart , Algorithms, Pseudo code (bubble sort, linear search - algorithms and pseudocode)

BASICS OF COMPUTER HARDWARE AND SOFTWARE

Computer is an advanced electronic device that takes raw data as input from the user and processesthese data under the control of a set of instructions (called program stored in memory) and gives the result (output) and saves output for the future use.

Computer Architecture



The basic components of a modern digital computer are: Input Device, Output Device, Central Processing Unit (CPU), and memory.

Central Processing Unit (CPU)

CPU is the brain of the computer system. All major calculation and comparisons are made inside the CPU and it is also responsible for activation and controlling the operation of other units. This unit consists of two major components that are arithmetic logic unit (ALU) and control unit (CU).

- **Arithmetic Logic Unit (ALU)**

Arithmetic logic unit performs all arithmetic operations such as addition, subtraction, multiplication and division. It also uses logic operation for comparison.

- **Control Unit (CU)**

The control unit of a CPU controls the entire operation of the computer.

It also controls all devices such as memory, input/output devices connected to the CPU.

- Additionally, CPU also has a set of registers for temporary storage of data, instructions, addresses and intermediate results of calculation
- When the ALU, CU and the registers are all integrated on a single chip, it is called a **microprocessor** thereby capturing the entire CPU on a single chip

Input /Output Unit

The input/output unit consists of devices used to transmit information between the external world and computer memory. The information fed through the input unit is stored in computer's memory for processing and the final result stored in memory can be recorded or displayed on the output medium.

Examples for input devices: Mouse, Keyboard, Scanner, Joy Stick, Microphone,
OCR (Optical Character Reader), MICR (Magnetic Ink Character Reader)
Examples for output devices: Printer (Dot Matrix, Ink Jet, Laser, Line), Monitor

Memory Unit

The Memory unit refers to the area where instructions and data are stored in a computer. The processor reads instructions and data from the memory, executes them and writes the result back to it. Different forms of memory are used in a computer system. They vary in their size, speed and location. Anything and everything in a computer is stored in the form of bits known as binary language or machine language. 8 bits make up a byte. The total number of bytes that a memory chip can hold at a time is termed as its size or capacity. Information is stored and processed as a group of bytes, called computer word. The word length is specific for each processor.

- **The Registers** - Since the group of registers is in-built within the processor chip, they are the fastest accessible units of memory by the processor. They are highly expensive and hence are limited in number. Some registers are used to store data while some are reserved to store address. There are some general purpose registers as well. Size of the registers depends on the processor. ***Accumulator*** is a register found on all processors, which is mainly used to store the operand of an arithmetic operation. ***Program***

counter (PC) always stores the address of the next instruction to be executed by the processor.

- **The Cache Memory**: - It is a small piece of high speed volatile memory located closer to the processor for fast processing. Cache memory is made of high speed SRAMs. It acts as a buffer between the CPU and the RAM. It holds frequently accessed data and instructions so that they can be easily supplied to the CPU when requested again. Cache memory is used to reduce the average time to access data from the Main memory.

- **Primary Memory**: - The memory chips that the processor can directly access are referred to as primary memory. The three types of primary memory are ***RAM, ROM*** and ***CMOS***. RAM is volatile whereas ROM as CMOS are non-volatile memories. They are metal oxide semiconductor memory cells built on silicon based ICs.

- **Random Access Memory (RAM):** RAM is a volatile memory and loses its contents when the power is turned off. The circuit is so designed that any random location can be directly accessed without the need to scroll up or down the memory. The program to be executed is loaded into RAM from the non-volatile backup memory. The processor reads instructions and data from the RAM, executes them and writes data back to the RAM. Hence RAM is also known as the main memory of the computer. Everything from the RAM is copied to the non-volatile backup memory before the computer is turned off. The two main types of RAM are **Dynamic Ram (DRAM)** and **static RAM (SRAM).**

Each memory cell in a DRAM is made of one transistor and one MOS capacitor, which store one bit of data. However, this cell starts losing its charge and hence data can be retained for less than thousandth of a second. So it needs to be refreshed thousand times a second, which takes up processor time. However, due to small size of each cell, one DRAM can have large number of cells. Primary memory of most of the personal computers is made of DRAM.

Each cell in SRAM is made of several transistors in a cross coupled flip flop configuration that stores one bit. It retains its bit till the power supply is on and doesn't need to be refreshed like DRAM. It also has shorter read-write cycles as compared to DRAM. SRAM is used in specialized applications.

Read Only Memory (ROM): ROM is a non-volatile memory and hence retains its contents even when the computer is switched off. It is used for storing software that is rarely changed during the life of the system, sometimes known as firmware. Almost every computer comes with a small amount of ROM containing the ***boot firmware*** which is essential for the boot-up. ***BIOS*** is a firmware used to perform hardware initialization during the booting process and to provide runtime services for operating systems and programs. The BIOS firmware comes pre-installed on the ROM and it is the first software to run when the computer is powered on. ROM memories have gradually evolved from fixed read-only memories to memories that can be programmed and then re-programmed.

- ROM (Read Only Memory)
- PROM (Programmable Read Only Memory)
- EPROM (Erasable Programmable Read Only Memory).
- EEPROM (Electrically Erasable Programmable Read Only Memory)

- **Complementary Metal Oxide Semiconductor (CMOS):** CMOS is a technology used to produce integrated circuits, but the term is commonly used to refer to the CMOS memory on the mother board. It is known as non-volatile RAM (NVRAM). It is powered by an onboard battery and stores system setting configurations that are required at boot time . In case the battery runs out, the CMOS is reset and loses all custom settings including the system clock. The information needs to be restored after replacing the battery.

Secondary Memory: - Primary memory has limited storage capacity and is volatile. Secondary memory overcomes this limitation by providing permanent storage of data and in bulk quantity. They are the slowest and cheapest form of memory. It cannot be accessed directly by the CPU. Programs need to be moved to primary memory in order to be executed by the CPU. Secondary memory is also termed as external memory and refers to the various storage media on which a computer can store data and programs.

The secondary storage media can be fixed or removable. Fixed storage media is an internal storage medium like hard disk that is fixed inside the computer. Fixed storage devices are literally not fixed; obviously these can be removed from the system for repairing work, maintenance purpose, and also for upgrade etc. Technically, almost all of the data i.e. being processed on a computer system is stored on some type of a built-in fixed storage device. Storage medium that are portable and can be taken outside the computer are termed as removable storage media.

Following are the main types of storage media:

Magnetic Storage Media: Magnetic media is coated with a magnetic layer which is magnetized in clockwise or anticlockwise directions. When the disk moves, the head interprets the data stored at a specific location in binary 1s and 0s at reading. Examples: hard disks, floppy disks and magnetic tapes.

Floppy Disk: A floppy disk is a flexible disk with a magnetic coating on it. It is packaged inside a protective plastic envelope. These are one of the oldest types of portable storage devices that could store up to 1.44 MB of data but now they are not used due to very less memory storage.

Hard disk: A hard disk consists of one or more circular disks called platters which are mounted on a common spindle. Each surface of a platter is coated with a magnetic material. Both surfaces of each disk are capable of storing data except the top and bottom disk where only the inner surface is used. The information is recorded on the surface of the rotating disk by magnetic read/write heads. These heads are joined to a common arm known as access arm.

Optical Storage Media: In optical storage media information is stored and read using a laser beam. The data is stored as a spiral pattern of pits and ridges denoting binary 0 and binary 1. Examples: CDs and DVDs

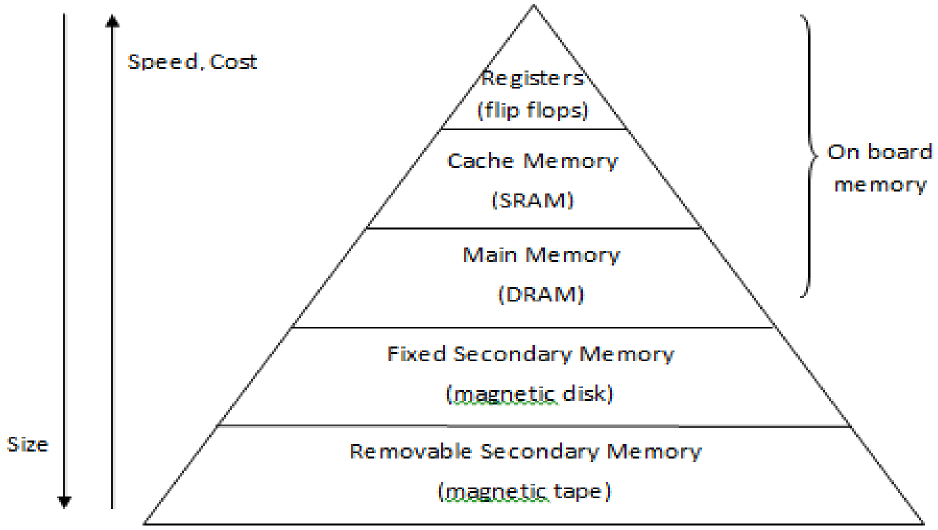
Compact Disk: A Compact Disc drive (CDD) is a device that a computer uses to read data that is encoded digitally on a compact disc (CD). A CD drive can be installed inside a computer's compartment, provided with an opening for easier disc tray access or it can be used by a peripheral device connected to one of the ports provided in the computer system. A compact disk or CD can store approximately 650 to 700 megabytes of data. A computer should possess a CD Drive to read the CDs.

DVD: It stands for Digital Versatile Disk or Digital Video Disk. It looks just like a CD and use a similar technology as that of the CDs but allows tracks to be spaced closely enough to store data that is more than six times the CD's capacity. It is a significant advancement in portable storage technology. A DVD holds 4.7 GB to 17 GB of data.

Blue Ray Disk: This is the latest optical storage media to store high definition audio and video. It is similar to a CD or DVD but can store up to 27 GB of data on a single layer disk and up to 54 GB of data on a dual layer disk. While CDs or DVDs use red laser beam, the blue ray disk uses a blue laser to read/write data on a disk

Solid State Memories: Solid-state storage devices are based on electronic circuits with no moving parts like the reels of tape, spinning discs etc. Solid-state storage devices use special memories called flash memory to store data. Solid state drive (or flash memory) is used mainly in digital cameras, pen drives or USB flash drives.

Pen Drives: Pen Drives or Flash drives are the recently emerged portable storage media. It is an EEPROM based flash memory which can be repeatedly erased and written using electric signals. This memory is accompanied with a USB connector which enables the pen drive to connect to the computer. They have a capacity smaller than a hard disk but greater than a CD.



Memory Hierarchy

HARDWARE

Hardware consists of the mechanical parts that make up the computer as a machine. The hardware consists of physical devices of the computer. The devices are required for input, output, storage and processing of the data.

Keyboard, monitor, hard disk drive, floppy disk drive, printer, processor and motherboard are some of the hardware devices.

SOFTWARE

Software is a set of instructions that tells the computer about the tasks to be performed and how these tasks are to be performed. Program is a set of instructions, written in a language understood by the computer, to perform a specific task. A set of programs and documents are collectively called software.

Software is classified into two categories

- System Software
- Application Software

SYSTEM SOFTWARE AND APPLICATION SOFTWARE

SYSTEM SOFTWARE

System Software is the type of software which is the interface between application software and computer hardware. Low level languages are used to write the system software. System Software maintains the system resources and give the path for application software to run. An important thing is that without system software, system cannot run. It is a ***general purpose software.***

Example: Operating System

Operating System

OS is a system software which acts as an interface between the user of a computer and computer hardware. It provides an environment within which the user can execute his programs.

Examples: Microsoft Disk Operating System (MS-DOS), Windows 7, Windows XP, Linux, UNIX, and Mac OS X Snow Leopard.

APPLICATION SOFTWARE

The software that a user uses for accomplishing a specific task is the application software. Application software may be a single program or a set of programs. It runs on the platform which is provided by system software. High level languages are used to write the application software. It is a ***specific purpose software***.

Examples

- Word Processing Software: For writing letter, reports, documents etc. (e.g. MS-WORD).
- Image Processing Software: For assisting in drawing and manipulating graphics (e.g. Adobe Photoshop).

The main difference between System Software and Application Software is that without system software, system cannot run on the other hand without application software, system always runs.

SL. NO	SYSTEM SOFTWARE	APPLICATION SOFTWARE
1	System Software maintains the system resources and give the path for application software to run	Application software is built for specific tasks.
2	Low level languages are used to write the system software.	High level languages are used to write the application software.
3	Machine Dependent	Machine independent
4	It is a general-purpose software.	It is a specific purpose software.
5	Without system software, system can'trun.	Without application software system always runs.
6	System software runs when system is turned on and stop when system is turnedoff.	While application software runsas per the user's request.
7	Compiler, Operating System, Interpreter	Photoshop, Microsoft Office

TYPES OF LANGUAGES

Computer programs can be written in high and low level languages, depending on the task and the hardware being used.

LOW LEVEL LANGUAGE (MACHINE LANGUAGE)

A program written in machine language is a collection of binary digits or bits (0 and 1) that the computer reads and interprets.

Features

- The computer can understand the programs written in machine language directly. No translation of the program is needed.
- Program written in machine language can be executed very fast (Since no translation is required).
- Machine language is defined by the hardware of a computer. It depends on the type of the processor or processor family that the computer uses, and is thus machine-dependent.
- A machine- level program written on one computer may not work on another computer with a different processor.
- It is difficult to write a program in machine language as it has to be written in binary code. Such programs are also difficult to modify
- Error occurring chance is high

ASSEMBLY LANGUAGE

A program written in assembly language uses symbolic representation of machine codes needed to program a particular processor (CPU) or processor family. This representation is usually defined by the CPU manufacturer, and is based on abbreviations (called mnemonics) that help the programmer remember individual instructions, registers, etc. Small, English-like representation is used to write the program in assembly language

Features

- Assembly language programs are easier to write than the machine language programs, since assembly language programs use short, English-like representation of machine code.

For example

```
ADD 2, 3
LOAD A
SUB A, B
```

- The program written in assembly language is the source code, which has to be converted into machine code, also called object code, using translator software, namely, assembler.
- Each line of the assembly language program is converted into one or more lines of machine code. Hence assembly language programs are also machine-dependent.

HIGH LEVEL LANGUAGE

High level languages are written in a form that is close to human language, enabling the programmer to just focus on the problem being solved.

Features

- Programs are easier to write, read or understand in high-level languages
- Programs written in high-level languages is the source code which is converted into the object code (machine code) using translator software like interpreter or compiler.
- A line of code in high-level program may correspond to more than one line of machine code.
- Programs written in high-level languages are easily portable from one computer to another.

Example: C, C++, Java, python

SYSTEM TRANSLATORS

A translator is a programming language processor that converts a computer program from one language to another. It takes a program written in source code and converts it into machine code. It discovers and identifies the error during translation. There are 3 different types of translators as follows:

- **COMPILER**

A compiler is a translator used to convert high-level language to machine language.

Eg: gcc, javac, g++

- **INTERPRETER**

Interpreter is a translator used to convert high-level programming language to low-level programming language.
Example: Python, jvm(Java Virtual Machine)

- **ASSEMBLER**

An assembler is a translator used to translate assembly language to machine language. Example: Fortran Assembly Program (FAP), Macro Assembly Program (MAP)

BASIS FOR COMPARISON	COMPILER	INTERPRETER
Input	It takes an entire program at a time.	It takes a single line of code or instruction at a time.
Output	It generates intermediate objectcode.	It does not produce any intermediate object code.
Working mechanism	The compilation is done beforeexecution.	Compilation and execution takeplace simultaneously.
Speed	Comparatively faster	Slower
Errors	Display all errors after compilation, all at the same time.	Displays error of each line one by one.
Error detection	Difficult	Easier comparatively
Example	Gcc, g++, javac	Python, jvm

STRUCTURED APPROACH TO PROGRAMMING

Structured Programming Approach can be defined as a programming approach in which the program is made as a single structure. It means that the code will be executed 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. Therefore, 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

Advantages of Structured Programming Approach

1. Easier to read and understand
2. Easier to Maintain
3. Easier to Debug
4. Machine-Independent, mostly.

FLOWCHART, ALGORITHM AND PSEUDO CODE

ALGORITHM

It is a complete step by step representation of the solution of the problem, represented in English like Languages. An algorithm can be abstract or quite detailed. A detailed algorithm consists of every step, equivalent to one instruction of a programming language.

Examples:

Algorithm to find the area of circle

1. Start
2. Read radius
3. area=3.14*radius*radius
4. Print area
5. Stop

Algorithm to find the sum of two numbers

1. Start
2. Read a, b
3. Sum=a+b

4. Print Sum
5. Stop

Algorithm to find the greatest of three given numbers

1. Start
2. Read three numbers to a,b and c
3. Check whether 'a' is greater than 'b'; if so, goto step 5 (Else control automatically goes to the step 4)
4. Check whether 'b' is greater than 'c'; if so display 'b' as the greatest number and goto step 5 ; else display 'c' as the greatest number and goto step 5
5. Check whether 'a' is greater than 'c'; if so display 'a' as the greatest number else display 'c' as the greatest number
6. Stop

Algorithm to find the average of N numbers

1. Start
2. Sum=0, Count=0
3. Read N
4. Read Num
5. Sum=Sum+Num
6. Count=Count+1
7. If Count<N , then goto step 4
8. Average=Sum/N
9. Print Average
10. Stop

Algorithm to find the largest of N numbers

1. Start
2. Count=0
3. Read N
4. Read Num
5. Large=Num

```
6. Read Num
7. If Num>Large, then Large=Num
8. Count=Count+1
9. If Count<N-1, then goto step 6
10. Print Large
11. Stop
```

Algorithm for bubble sort

```
1. Start
2. Get the total number of elements of the array in variable, ‘n’
3. Read the numbers into array a
4. Let i=0
5. Let j=0
6. Compare jth number with the (j+1)th number; if it is greater, swap them
7. Increment j
8. Repeat step 6 & 7 until j= n-i-1
9. Increment i
10. Repeat steps 5 to 9 until i=n-1
11. Print the sorted array
12. Stop
```

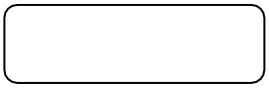
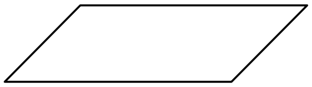
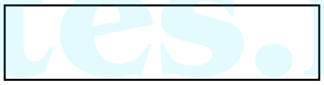
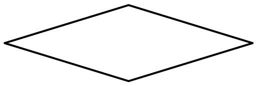
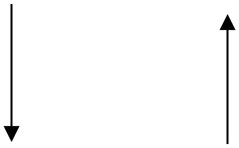
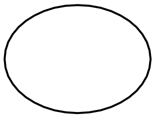
Algorithm for linear search

```
1. Start
2. Get the total number of elements of the array in variable, ‘n’
3. Read the numbers into array a
4. Read the number to be searched in x
5. Set flag as 0
6. Let i=0
7. Check whether ith element of a is equal to x;  if yes, set flag as 1 and goto step 10
8. Increment i
9. Repeat step 7 & 8 until i= n-1
10. If flag is 1, print “the element is found at position i+1”; else print “element not found”
```

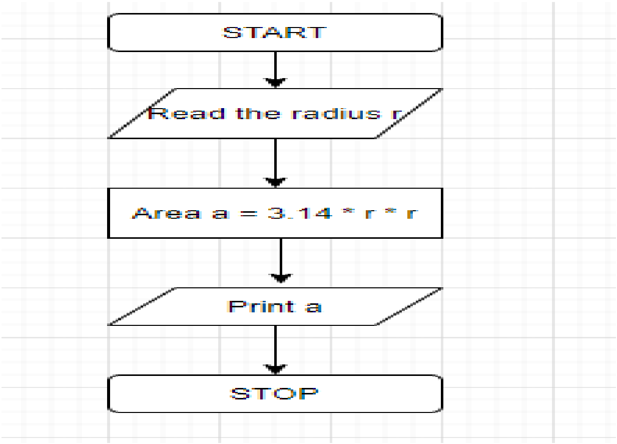
11. Stop

FLOWCHART

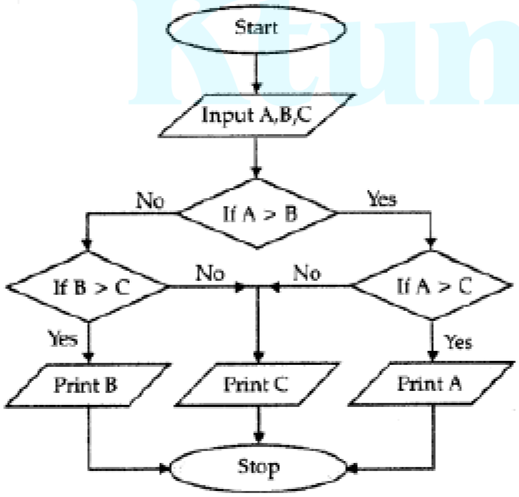
A flowchart is a diagram that visually represents the steps involved in solving a problem or carrying out a process. It depicts the possible ways along which the control can flow. A flowchart can also be defined as a diagrammatic representation of an algorithm. Flowchart uses many graphical symbols and thus, is more understandable. The symbol used for different types of statements are as shown

Start, Stop	
Read, Print	
Processing Statements	
Condition Check	
Direction of flow	
Connectors (for longer flow chart)	

Example: ***Flowchart for finding the area of a circle***



Flowchart to find the greatest of three numbers



PSEUDO CODE

It is a more formal representation than the algorithm. Here, we represent every step in a formal way which is very close to the actual programming language representation. In pseudocode, each of the steps will be written via operator and statements equivalent to some programming language instructions. The only difference will be that the exact syntax of the programming language will not be followed. All pseudocodes will start with the keyword "START" and complete with keyword "STOP" or "END".

Example: ***Pseudocode to find the area of circle***

```
1. START
2. Read radius
3. area = 3.14 * radius * radius
4. print area
5. STOP
```

Pseudocode to find the greatest of three numbers

```
int a, b, c
print: enter the numbers
scan:  a, b, c
if a>b
    if a>c
        print: a is the greatest   /* a>b and a>c, so a is the greatest
    else
        print: c is the greatest    /* a>b but a<c which implies c is the greatest */
elseif b>c
        print: b is the greatest    /* a<b and b>c then b is the greatest */
    else
        print: c is the greatest    /* a<b and b<c which implies c is the greatest */
```

SAMPLE QUESTIONS

1. Write short note on processor and memory in a computer. ***[KTU MODEL 2020]***
2. What are the differences between compiled and interpreted languages? Give example for

each. ***[KTU MODEL 2020]***

3. Write a pseudo code representing the flowchart for linear searching. ***[KTU MODEL 2020]***
4. Write an algorithm representing the flowchart for bubble sort. ***[KTU MODEL 2020]***
5. Draw a flow chart to find the position of an element in a given sequence, using linear searching technique. With an example explain how the flowchart finds the position of a given element. ***[KTU MODEL 2020]***
6. With the help of a flow chart, explain the bubble sort operation. Illustrate with an example. ***[KTU MODEL 2020]***
7. What is cache memory? ***[KTU, DECEMBER 2019]***
8. What is an algorithm? Give the basic properties of an algorithm. ***[KTU, DECEMBER 2019], [KTU,MAY 2019]***
9. Draw a flowchart to find the factorial of a number. ***[KTU, DECEMBER 2019], [KTU, APRIL 2018]***
10. What is an operating system? Explain any three types of OS? ***[KTU, DECEMBER 2019]***
11. What are the key features of an interpreter? ***[KTU, DECEMBER 2019]***
12. Give an algorithm and flowchart to generate prime numbers in a given range. ***[KTU, DECEMBER 2019]***
13. Differentiate system software and application software. ***[KTU, MAY 2019], [KTU, JANUARY 2017]***
14. Draw a flowchart to find area of a triangle. ***[KTU, MAY 2019]***
15. Formulate an algorithm and draw a flowchart to generate Fibonacci series up to n term***s. [KTU, MAY 2019]***
16. What is high level language? Give four examples***. [KTU, DECEMBER 2018]***
17. Write an algorithm to check whether a number is odd or even***. [KTU, DECEMBER 2018]***
18. What is memory hierarchy? Explain with a neat diagram. Compare in terms of speed, cost and storage. ***[KTU, DECEMBER 2018]***
19. Give an algorithm and flow chart to find the largest among N number*s* ***[KTU, DECEMBER 2018]***
20. What is a flowchart? What are the notations used in flowcharts? ***[KTU, APRIL 2018], [KTU, JULY 2017]***
21. Write an algorithm and draw a neat flowchart to find all the possible roots of a quadratic

equation*.* ***[KTU, DECEMBER 2017]***

22. Explain in detail the difference between procedural programming and object oriented programming***. [KTU, APRIL 2018], [KTU, MAY 2019]***
23. Draw the flowchart to print the largest among two integers. ***[KTU, JUNE 2017]***
24. Develop an algorithm to swap two integers without using temporary variable***. [KTU, JUNE 2017]***
25. What is an algorithm? Write an algorithm for reverse a number. ***[KTU, DECEMBER 2018]***
26. How are programming languages classified? ***[KTU, JULY 2017]***
27. Write an algorithm and program which accepts a number from the user and print whether it is a prime number or not. ***[KTU, APRIL 2018], [KTU, DECEMBER 2018]***
28. Draw the flowchart to generate the numbers between 100 and 200 which are divisible by 3, but not divisible by 4. ***[KTU, JANUARY 2017]***
29. Write an algorithm to count the number of digits in a positive integer. ***[KTU, JANUARY 2017]***
30. Give the algorithm and flowchart for finding the largest and smallest numbers in a given list of N numbers ***[KTU, JANUARY 2016]***
31. Write an algorithm to compute sum of the series $1-\frac{x^2}{2}+\frac{x^4}{4}-\frac{x^6}{6}+\cdots n\ terms$***[KTU, JANUARY 2016]***
32. Give the algorithm and draw flowchart for generating Armstrong number between the given ranges. (Armstrong number is a number z such that sum of cubes of the digits of z is equal to n) ***[KTU,JANUARY 2017]***
33. Draw the flowchart to find out the greatest of three number*s. **[KTU, JANUARY 2016]***
34. Write the algorithm to perform bubble sorting and illustrate the step by step changes in the following array of numbers for each iteration.

24	***14***	***36***	***79***	***88***	***12***	***6***

[KTU, JUNE 2017]