IBPS SPECIALIST OFFICER STUDY MATERIAL (IT SCALE-I)

Data Base Management Systems (Introduction)

Data : Data is the raw material from which Useful Information is derived. The word data is the Plural form of Datum (but nowadays, people commonly using data for both singular and plural).

Simply we can say that, data is a collection of unorganized facts, but can be made organized into useful information.

Some examples of data are Prices, Weights, Costs, Number of items sold or purchased etc.

Information : The data, that have been processed in such a ways so as to increase the knowledge of the person who uses the data is known as information. Of-course, now a days these terms data and information are closely related and now a days people often interchanging their meanings.

Data (or) Information Processing : The process of converting the facts into meaningful information is known as Data processing. It is also known as Information processing.

Meta Data : Simply we can say it as the Data about the Data. In other words, The Data that describes the properties of other data is known as Metadata. I mean, Meta Data keeps the information of the Other Data, i.e., it keeps the information HOW and WHERE the other data is stored.

Data Base : A database is a collection of data (information) of some given Organization (a company for example), that can be processed through one or more programs by multiple users.

Some examples are A bank, A hospital, A company etc.....

Data Base Management Systems :

Now you can define the database management systems easily, A Data Base Management System (simply called as DBMS) is A System that manages the Data Base.....

Not Satisfied? ok " A Software that manages the collection of Data"

Need a technical Definition? "**A software that provides an efficient environment to help us in STORING and RETRIEVING the Data Base of an enterprise.**

Our DBMS, Defines Data, Stores Data, Maintains data and ofcourse, Deletes the Data...
Simply our Database is the CONTAINER for a collection of computerized data files. Our DBMS tells us n helps us in how to insert, maintain and delete these files.

Now lets have a look at some advantages of Database Management Systems

- **Reduction of Redundancies :** Redundancy means duplicating (I mean, making the same copy of data again and again). Reduction of Redundancy means, avoiding the duplication of data (Remember the warning message "The file named XYZ.JPG is already exist. do you want to replace it???", yea the same thing
- **Data Independence and Efficient Access :** The files stored in Database are independent of their storage details. And the change in one file doesn't effect the other (until and unless they related to each other).
- **Data Integrity :** Data Integrity means, that the data values entered in the database must be checked to ensure that they fall within the correct format and range. Confused? lets see an example. Suppose I am entering the students details of a School. what is I entered the age of the student as 55 ??? people will laugh at me. So the value should be less than 17. And of course, the Data Integrity also checks whether the referring field is existing or not. I mean, if i want to enter the marks of a person named shivani sharma, first of all it checks whether the name of that person exists in its database or not...
- **Data Security :** All of us knows about Security. So no need to discuss. We shall discuss about this in detail later...
- **Less Application Development Time :** With the help of some Predefined functions like Concurrency control, Crash Recovery etc, DBMS helps us to develop applications in very less time.

- **Conflict Resolution :** It helps us in resolving the conflicts among various users to access the same data file
- **Data Administration :** DBMS provides maintenance and administration of data by providing a common base for the large collection of data being shared by several users.
- **Concurrent Access :** Number of users can access a single file Concurrently (I mean, at the same time).
- **Crash Recovery :** The DBMS maintains a continuous record for the changes made to the data, so , if there is any system crash by power failure or something, it can restore the Database...

The Transaction Management

Assume that you want to transfer some money (Say Rs. 1500) to your friend's account....

You logged in into your bank's site and types your account number.

Later you typed your friend's name and his account number

Later types the amount and clicks on TRANSFER button.

Later??? Rs. 1500 /- TRANSFERS into your friend's account.

Now lets see this process in DBMS's point of view.

You entered some money (Rs 1500). The DBMS checks whether the mentioned money is available in your account or not .Assume that you have some 3000 in your account.

If available then it checks the reciepient's Account (Assume that he has Rs. 200 in his account)

Later the DBMS reduces your amount to Rs 1500 and Adds that 1500 to your friend's account.
And makes his account balance as Rs 1700 (200 + 1500).

Then shows you a message, something like "TRANSACTION COMPLETED"

yes, this is an example of Transaction.

A transaction is an execution of a user program and is seen by the DBMS as a series or list of actions. (or simply we can remember it as " A transaction is nothing but a List of Actions". These actions include the reading and writing of database.

ACID Properties (important) :

These are the properties that a transaction should possess in order to avoid failures during concurrent access to a database. The **ACID** is an acronym which stands for **A**tomicity, **C**onsistency, **I**solation, **D**urability. Now lets have a look at these properties in detail. Of course, for beginners these all may look same and confusing, read them two or three times. then you can get the exact meanings and differences among them...

- **Atomicity :** It ensures that the transaction either is executed **completely** or not at all. **I**ncomplete transaction consequences are not entertained, check an example
    - Assume that Shivani has Rs. 500/- in her account and Palvi has Rs. 200/- in her account. Now Shivani transfers an amount of Rs. 50/- to Palvi. A transaction debits the amount from shivani's account, but before it could be credited to palvi, if there is a failure, then transaction would stop. So finally Shivani loses Rs. 5o but palvi cant get the amount. This leaves the data in an inconsistent state. If there is a failure during transaction execution , then measures must be taken to get back the data in a form which was in, before transaction (I mean, the 50 shouldn't be deducted from Shivani's account in our case). This is taken care of by transaction management component.
- **Consistency :** The data in the database must always be in a consistent state. A transaction occurred on a Consistent data should end with the data with another Consistent stage after completion of that transaction. Take the above case, the total of the amounts of Shivani and Palvi are (500+200) is Rs 700/- So, after the Transaction completed, the total amount should be same. i.e., (450+250 = 700). Of course, in intermediate stage, where the amount is deducted from Shivani's account but not yet credited to Palvi, the total would not be same. It is the responsibility of DBMS.
- **Durability :** Durability ensures that the data remains in a consistent state even after the FAILURE. (This is ensured by keeping copy of the old data in the Disk, till the transaction is COMPLETED). I mean, if shivani is transferring money to Palvi. The money is deducted from Shivani's account

and power gone (before adding the MONEY to palvi's account). Then our DBMS shouldn't save that transaction. This is called Durability.

- **Isolation :** All transaction must run in Isolation from one another. I mean, each and every transaction should be kept unaware of other transactions and execute independently. The intermediate results shouldn’t be available to other transactions.

Data Base Users and Administrator

Depending on their degree of expertise or the mode of their interactions with the DBMS, The Data Base users (people who uses database) can be classified into several groups.
They are,

- Naive Users
- Online Users
- Application Users
- Sophisticated Users
- Specialized Users

Naive Users : Naive means Lacking Experience, these are the users who need not be aware of the presence of the Data Base System. Example of these type of users is The user of an ATM machine. Because these users only responds to the instructions displayed on the screen (enter your pin number, click here, enter the required money etc). Obviously operations performed by these users are very limited.

Online Users : These are the users who may communicate with the Data Base directly via an online terminal or indirectly via a user interface and application program. These users are aware of the presence of the Data Base System and may have acquired a certain amount of expertise within the limited interaction they are permitted with a Data Base.

Application Programmers : Professional / Application programmers are those who are responsible for developing application programs or user interface. The application programs could be written in a general-purpose programming language or the commands available to manipulate a database.

Sophisticated Users : Simply we can say that these are the EXPERIENCED users. These people interact with the system without writing programs. Instead they from their requests in a database query language. They submit each such query to a query processor, whose function is to break down DML (Data Manipulation Language, the language which is used to MAINTAIN the data. we shall discuss about this later) statements into instructions that the storage manager understands. Analysts who submit queries to explore data in the Data Base fall in this category.

Specialized Users : These are the sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework. Among these applications are computer - aided design systems, knowledge-based and expert systems, systems that store data with complex data types (Ex, Graphics Data and Audio Data) and environment-modelling systems.

DBA (Data Base Administrator) : The person who controls both Data and the Programs that access that data in the Data Base is called the Data Base Administrator (DBA).

Functions of the DBA are,

- Defining Schemas (arrangement of Data)
- Creating Storage Structure and Access Methods (I mean how to store data and access that)
- Modifying the storage Data
- Granting Authorization Permissions
- Specifying the CONDITIONS of the data storage
- Periodically Updating the Data Base etc.,

Data Models

Data Model means, to give a SHAPE to the data. A Data Model Makes it easier to understand the Data. We can define the data model as "The Collection of High-Level data description that hide many low level storage details".
The Data Models are divided into THREE different groups. They are,

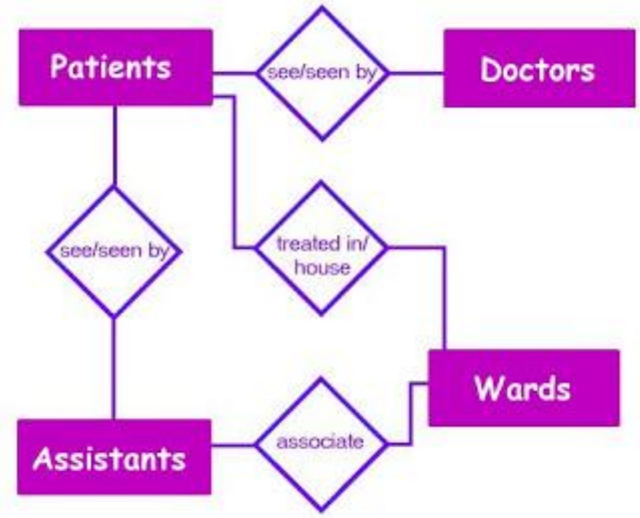
1. Object - Based Logical Models
2. Record - Based Logical Models
3. Physical Data Models

1. **Object Based Logical Models :** These are used in describing data at Logical Level and View Level. (Logical level describes **what data are stored in the database** and **what relationships exists among** those Data. This logical level is used by the DBA (Data Base Administrator). He must decide what information is to be kept in the Data Base. The View level describes Only **part** of the entire database to be viewed by the user of the database hiding the details of the Information Stored.

The Object based logical models are described in the different following models.

- The E-R (Entity-Relationship) Model
- The Object-Based Logical Model
- The Semantic Data Model
- The Functional Data Model

E-R Model : The entity is a "Thing" or " Object" in the real world that is distinguishable from other objects. The E-R model is based on the collection of basic objects called **Entities** and the **Relationship** among them. Consider the following Diagram.



In the above diagram, RECTANGLES represents ENTITIES, DIAMONDS represents RELATIONSHIP among those ENTITIES. LINES represents links of Entities to Relationships.

Object - Oriented Model : The Object Oriented model based on a collection of OBJECTS. An object contains values stored in Instance Variables and Bodies of Code that operates on the Object. (These bodies of Code is called Methods).
Objects that contain the same types of values and the same methods are grouped together into classes. (A class is the definition of the object).

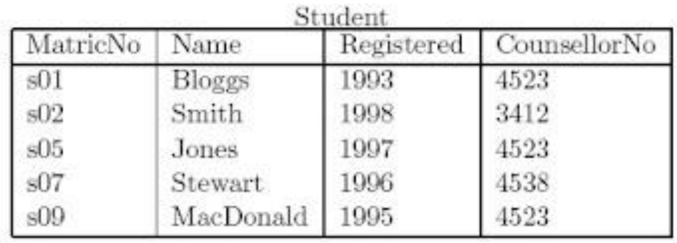
Semantic Data Model : A Semantic data model is a more high level data model that makes it easier for a user to give **Starting Description** of the data in an organization. (Semantic is nothing but the Meaning). These models contain a wide verity of relations that helps to describe a real application scenario. A DBMS cannot support all these relations directly. So it is build only with few relations known as relational model in DBMS. A widely used semantic data model is the Entity-Relationship (ER) data model which allows us to graphically denote entities and relationship between them.

Functional Data Model : The functional data model makes it easier to define functions and call them where ever necessary to process data.

2. **Record - Based Logical Models :**
In this type of models, the data is kept in the form of RECORDS (documents). Thse models describes data at Logical and View Levels. When compared with Object Based Data Models, the record based logical models specifies the overall logical structure of the database and provides higher-level implementation.
These are of 3 types. Those are,

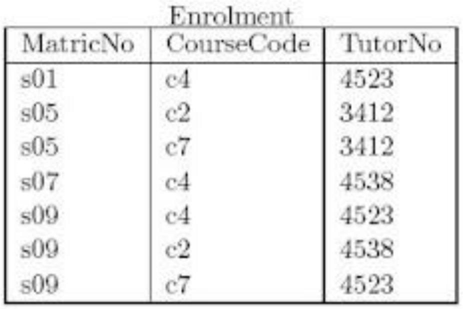
- Relational Model
- Network Model
- Hierarchical Model

Relational Model : The relational model represents both Data (entities) and Relationships among that data in the form of **Tables.** Each table contains multiple columns and each column contains a unique name. Have a look at the following table.



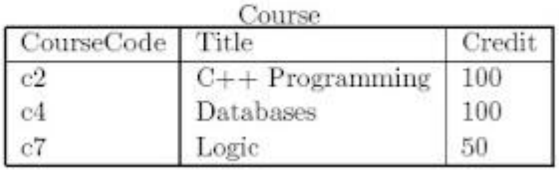
Student

MatricNo	Name	Registered	CounsellorNo
s01	Bloggs	1993	4523
s02	Smith	1998	3412
s05	Jones	1997	4523
s07	Stewart	1996	4538
s09	MacDonald	1995	4523



Enrolment

MatricNo	CourseCode	TutorNo
s01	c4	4523
s05	c2	3412
s05	c7	3412
s07	c4	4538
s09	c4	4523
s09	c2	4538
s09	c7	4523



Course

CourseCode	Title	Credit
c2	C++ Programming	100
c4	Databases	100
c7	Logic	50

Network Model : Data in the network model are represented by collection of Records and Relationships among data are connected by **LINKS.** These links can be viewed as Pointers..

Hierarchical Model : This is also same as Hierarchical model, the difference is the records in the database are represented in the form of TREES (in Hierarchical way)

Physical Data Models: Physical data models are used to describe data at the lowest level, which explains how the data is actually stored using complex low-level data structures.

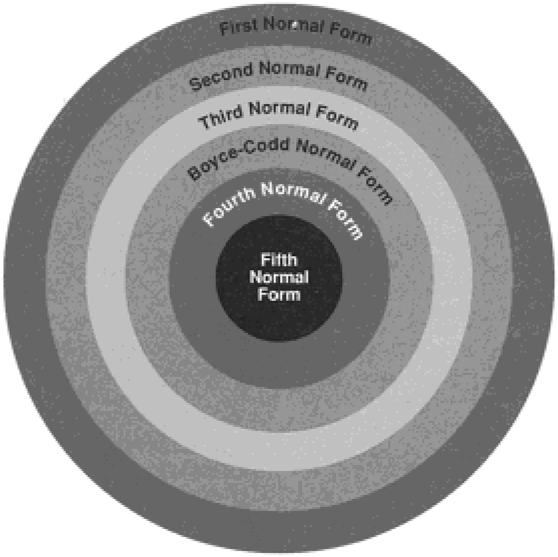
There are two types in Physical Data Models.

- Unifying Model
- Frame-Memory Model

Normalization and Normal Forms

The Process of making a table (or often we call it as Record) NORMAL is called Normalization : Ok in other words, we can say that the Normalization is the process of simplifying the Table :)

The design method which helps us to MINIMIZE the redundancies of data (repetitions) and reduces the errors / mistakes in the table is called Normalization.



Note : To perform this Normalization Process, we should analyze the INTER DEPENDENCIES of the attributes in the tables and should take the subsets of the larger tables to form the Smaller tables.

In this process, we apply various **Normal Forms** to the given Database (Table). These Normal Forms helps us to break the BIG tables into smaller ones. Now lets have a look at these Normal Forms...

1st Normal Form :

***** In 1st Normal Form we deal with the REDUNDANCY of ROWS.
In first Normal Form, each ATTRIBUTE must be ATOMIC. There shouldn't be any repetitions in the Rows. and There shouldn't be any multivalued columns.

The advantage of the 1st Normal Form is "**The queries become easier"**

The 1NF eliminates GROUPS by keeping them into different (separate) Tables and connecting them with One to Many relationship.

2nd Normal Form :

***** A table or entity should be in the first normal form, if you want to apply the 2nd Normal form on that.

Note : The first normal form deals with the repetition (redundancy) of data in Rows (Horizontal) , where as the 2nd Normal form deals with the redundancy of the data in Columns (Vertical).

As I already said, A table should be in the First Normal form if you want to make sure that it is in 2nd normal form. Some more additional conditions are,

- It shouldn't have a COMPOSITE primary key. (I mean, the key SHOULD NOT be divided into sub parts (logical entities).
- In 2nd Normal Form, each attribute should FUNCTIONALLY Dependent on the primary key.
  - **What is this Functionally Dependence?**
    - If one or more attributes determines UNIQUELY the values of other attributes, then that property is called as Functionally Dependence.
- If you find any Non Dependent Attributes, you should move them into Smaller Tables.
- 2NF helps us to improve the data Integrity (Error free)

3rd Normal Form :

***** A table or entity should be in the Second Normal Form (so obviously, it should be in the first normal form too :P) if you want to apply 3rd Normal form on that. Some more conditions are,

- If the table is in 3rd Normal Form, then it should be **Transitive dependencies** free.....
  - **What is this Transitive Dependence?**
    - If there exists TWO separate entities in ONE table. Obviously we should move them into smaller tables.
- 3nf also further improves the integrity of the data.

Boyce - Codd Normal Form : (Not Boys Code as some of us are thinking :P)

- It is the ADVANCED version of the 3rd Normal Form (3NF).

- A row in a table is in BCNF, if every determinant of it is a Candidate Key. (Most of the entities of 3rd Normal form are already in Boyce - Codd Normal Form)
    - **Why should we use BCNF if already 3NF is there?**
        - The 3rd Normal Form misses the inter dependencies between Non Key attributes.
        - Simple we can say that, a 3NF is not in BCNF if,
            - There are number of Candidate Keys
            - The keys are composed of multiple attributes
            - There are more than one attribute between the Keys.......

4th Normal Form :

Obviously it should be in the 3NF if you want to call it 4NF. In addition, we should eliminate Trivial Multi valued Dependencies... I mean, our table shouldn't have multiple sets of Multi Valued Dependencies.

5th Normal Form :
***** It should be in the 4th Normal Form...

- In this, we should eliminate the dependencies, not determined by the Keys.
- I mean, every dependency should be the consequence of its Candidate keys

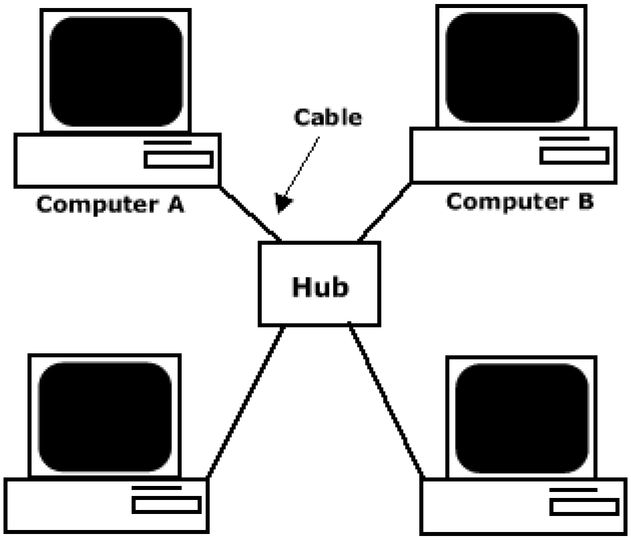
Data Communications and Networking

Computer Networks (Basics)

Suppose you want to chat with your friend, or want to share your printer with your friend's system. Then you should connect your computer to his/her computer. In simple words, you are just forming a NETWORK by connecting two or more computers. These connections or arrangements are called Computer Networks. These computers in the Computer Network may attached with cables, telephone lines, radio waves, satellites or infrared light beams.

There are two aspects in computer networks.

- **Hard Ware :** It includes physical connection (using adapter, cable, router, bridge etc)
- **Soft Ware :** It includes set of protocols (nothing but a set of rules)



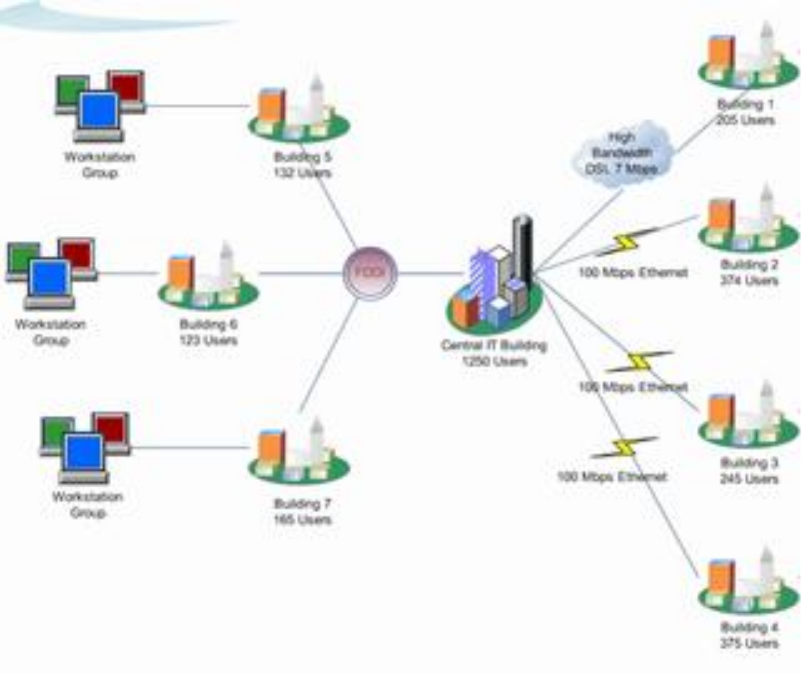
Basic types of Computer Networks :

Local Area Networks (LANs) :

These are the simple networks. A set of computers (2 or more) connected within a single building or college campus (a few kilometres range)

Metropolitan Area Networks(MANs) :

These are the larger networks than LANs, these are limited to city or group of Nearby Offices. Simply we can say that a group of LANs are called a MAN (but guys, this is not a standard definition, its jst for your understanding purpose).



Wide Area Networks (WANs) :

These type of networks provides long distance transmission of data and voice. (A city, state , country and ofcourse the world). it contains HOST and collection of machines.

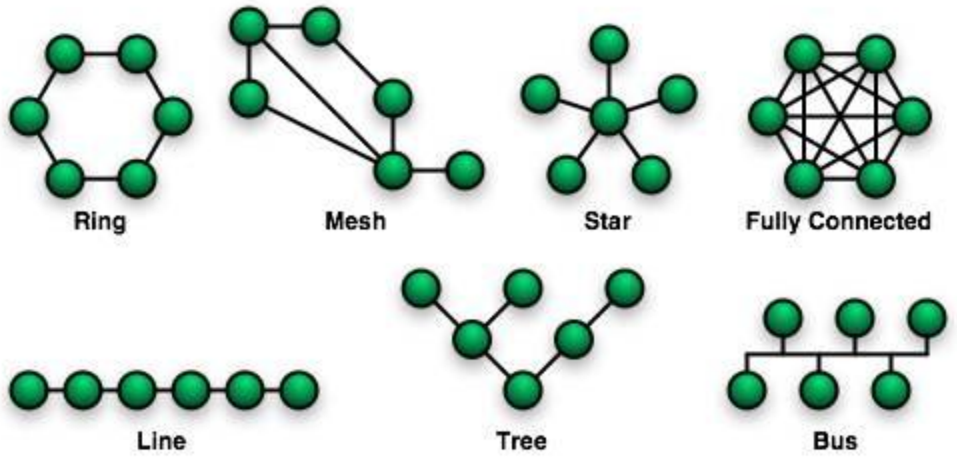
Now lets have a look at some basic differences between these three...

- **Area :**
  - **LAN :** Covers small area.
  - **WAN :** Covers large geographical area
  - MAN : Covers larger than LAN but smaller than WAN
- **Error Rates :**
  - LAN : Lowest
  - WAN : Highest
  - MAN : Moderate
- **Transmission speed :**
  - LAN : High Speed
  - WAn : Low Speed
  - MAN : Moderate speed
- **Cost :**
  - **LAN :** Uses inexpensive equipment (i mean, if you have two systems, its hardly costs nothing (almost) to connect them with each other)
  - WAN : Uses most expensive equipment (Connecting systems via internet)
  - MAN : Uses moderately expensive equipment (Connecting two systems in different buildings)

Till now we have discussed about the types of Networks. Now we shall discuss about HOW should we connect them.

Network Topology : A network Topology is the method or process which tells us about how to connect the Systems.

There are several types of Topologies are there.



Assume that the above green dots are computers.

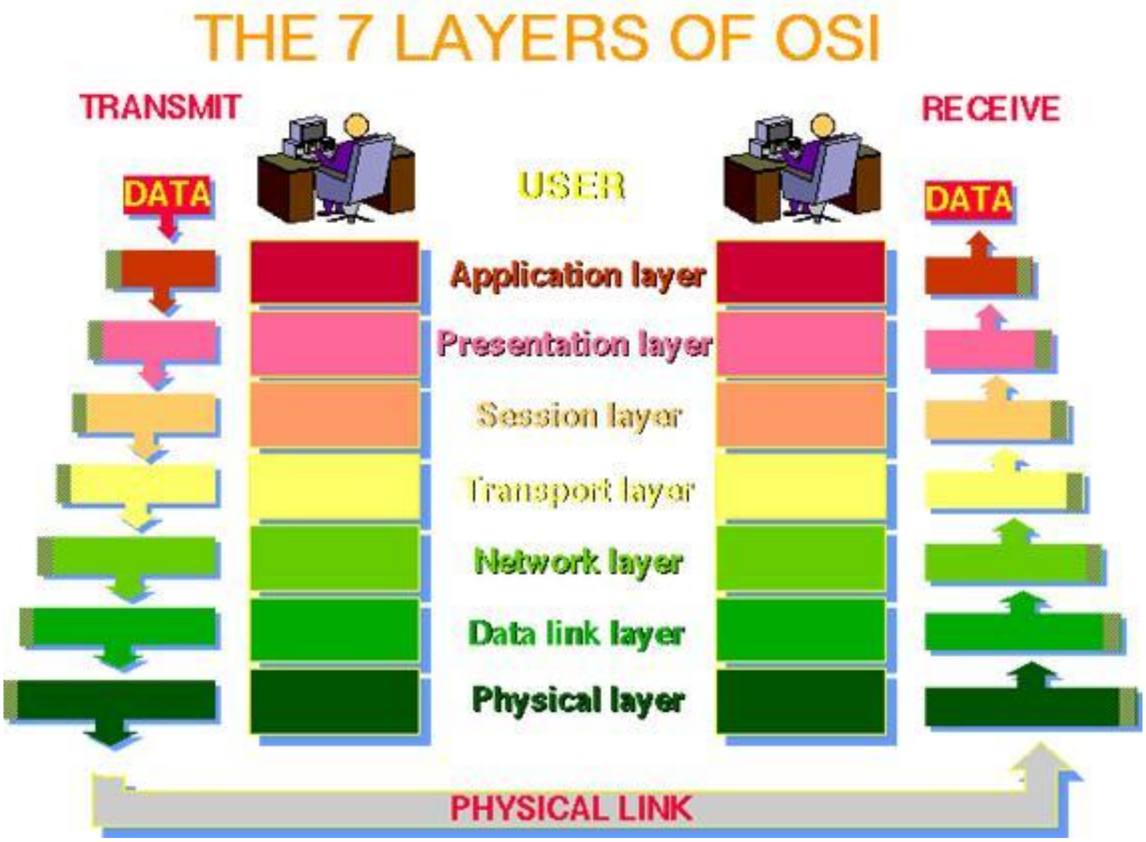
Line Topology : If we connect several computers in a line then it is called as the LINE TOPOLOGY
Treee Topology : If we connect several systems in a hierarchical order, then it is called as the TREE TOPOLOGY
Bus Topology : If we connect several computers to a back bone (A wire) then it is called as the BUS TOPOLOGY.
Star Topology : If we connect several computers to a central hub (system) then it is called as the STAR TOPOLOGY.
Ring Topology : In line topology if we connected the last system to the starting system, then it is called as the Ring topology.
Mesh Topology : If we connect several systems with each other (There is no necessity that one system should connect with all the remaining systems in the network, see the diagram) then it is called as the Mesh Topology.
Fully Connected : In a Network if everysystem is connected with all the remaining systems then it is called as the Fully Connected topology.

The OSI Model

In this model, similar types of communication functions are grouped into Layers. There are 7 layers in this Model. Those are,

1. Physical Layer
2. Data link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

Each layer helps it's above layer to establish an error free communication, in other words, each layer depends on its below layer to establish an error free communication.



How to remember these names????
Here the order is very important. If anybody asks you about these layers, you should start with Physical and end with Application Layer. You should not jumble these names according to your convenience. So you should remember these names

with an order. Well, there are several techniques are there to remember these type of names. One of those technique is, making a funny name or sentence with the STARTING letters. Just, write the starting letters of these layers.

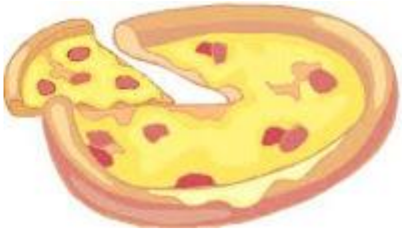
P D N T S P A

Now make a funny sentence, here are some examples

- Please Do Not Touch Steve's Pet Alligator

Or

- Please Do Not Throw Sausage Pizza Away

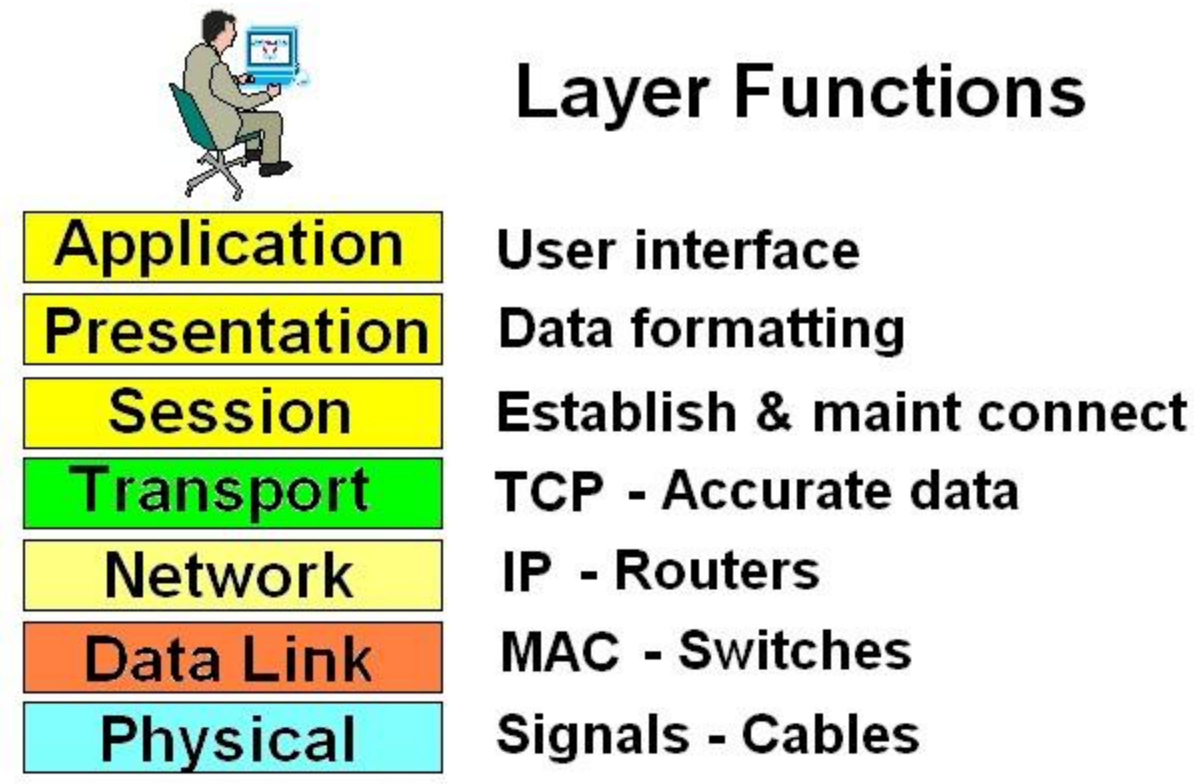


Now I bet you never going to forget these names and order again :)

Ok, now lets have a look at the Functionalities of these Layers...

- **Physical Layer :** This is the First or Lowest layer of the OSI model. It takes care about the Physical Connections.
- **Data Link Layer :** This layer Encodes the data packets and provides error free node to node transmission. It is divided into two sub layers. Those are
    - Media Access Control Layer (MAC Sublayer in short)
        - It helps the computer to get the access of data transmission
    - Logical Link Control Layer (LLC Sublayer in short)
        - It controls the order, flow and errors of the frames (Frame synchronization)
- **Network Layer :** This layer helps in Switching and Routing (connecting and transmitting), and also creates virtual circuits (Imaginary paths for transmission)
- **Transport Layer :** This layer helps in providing Transparent data transmission. Its also checks whether the data completely transmitted or not.
- **Session Layer :** This layer is responsible for the Transmission Sessions. I mean Starting, Maintaining and Terminating the Communication.
- **Presentation Layer :** This layer provides the data Independence (simply you can remember that this layer is responsible for the data Presentation :) I mean, how you are presenting data (without any leakage)
- **Application Layer :** This is the topmost layer, this layer directly interacts with the end user. This layer is responsible for checking whether the claimed user is genuine or not.

Still having problems with these functions??? Have a look at the following diagram. It will be easier for you to understand :)



Layer Functions

Application	User interface
Presentation	Data formatting
Session	Establish & maint connect
Transport	TCP - Accurate data
Network	IP - Routers
Data Link	MAC - Switches
Physical	Signals - Cables

Very good. Now you know about the OSI model and It's Layers :) You know that it was developed by ISO.

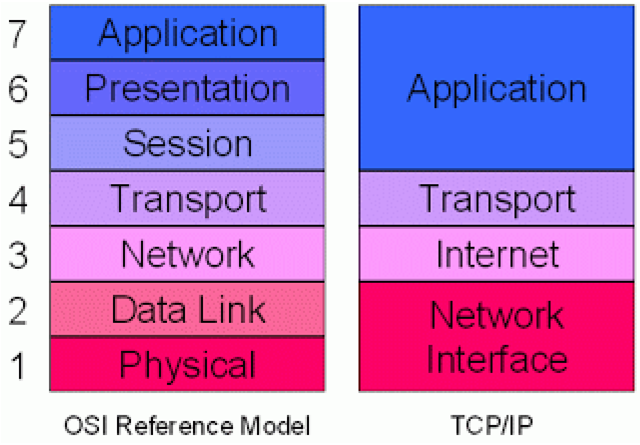
Hey, you know one secret? ISO dint invent any new model. It just COPIED some old model and modified it.. Thats all.

Once there was a model called TCP/IP model (Transport Control Protocol / Internet Protocol) with this name itself, you can define its functionality :) It has lil complicated structure with only **4 Layers.**

Those are

1. **Network Interface Layer**
2. **Internet Layer**
3. **Transport Layer**
4. **Application Layer**

ISO Just elaborated this model and divided some of these layers into EXTRA LAYERS and developed OSI model. have a look at the following Diagram, so that you can understand the difference :)



Compiler Design

Compilers Basics

Assume that you visited china to meet China's Prime minister. The meeting was arranged in a big auditorium. But now the problem is, you don't know Chinese and he too don't know your Language (say Hindi). Then what to Do? Now you have 3 solutions.

1. To learn Chinese
2. To make him learn your Language
3. Or you both should learn some other language (Like English) so that you can communicate in that language...

Now what if HE is not ready to learn your language or English??? and what if you too are Lazy enough to learn Chinese or English??? Isn't there any other solution for this problem???

There is...... If you have enough money, You can hire a person who knows your language as well as Chinese and he will work as a MEDIATOR for both of you people.. :)

Now come to our Subjects...... As we know, computer cant understand our **Hais and hellos....**
And we too cant understand Computer's **0's and 1's...** So how to talk to the computer and make it works according to your orders?

Can you teach English to computer? Its impossible, because its a machine. It cant understand anything except LOW VOLTAGE (0) and HIGH VOLTAGE (1)... On and Off in simple words...

We too are TALLENTED enough to use CTRL+C and CTRL+V so we are not ready to write big codes which looks like 00011 01001 110100 1001001 01001 11001 11111 0111010 101010 10101011 01111 1101010 11101 1010101 01010 n bla bla........

So what can we do now? we write a **Software Code** which acts as a TRANSLATOR and converts your instructions into 0's and 1's and gives them to computer and vice versa.. We call this SOFTWARE CODE as **COMPILER..** Thats it :)

What exactly is the Compiler? is it a for Translating or for Debugging?
Lets go back to our Chinese Man, assume that you are communicating with him with the help of a Translator. And you have some problem with your TONGUE... (assume)..... So, what will happn if you use some adjectives like "Poker face and ######## (censored :P) " .
Now say, what if the Mediator translates that POKER FACE stuff as it is in Chinese???
Your relationship with the Chinese man, will collapsed and you will get some injuries if he is expert in those Chinese martial arts...

So, before going to translate your words in Chinese, your mediator (or Translator) warns you about the outcomes of your words (may be you will get kicked or killed, depends upon the intensity of your words :P) and confirms whether to translate them as they are or not...

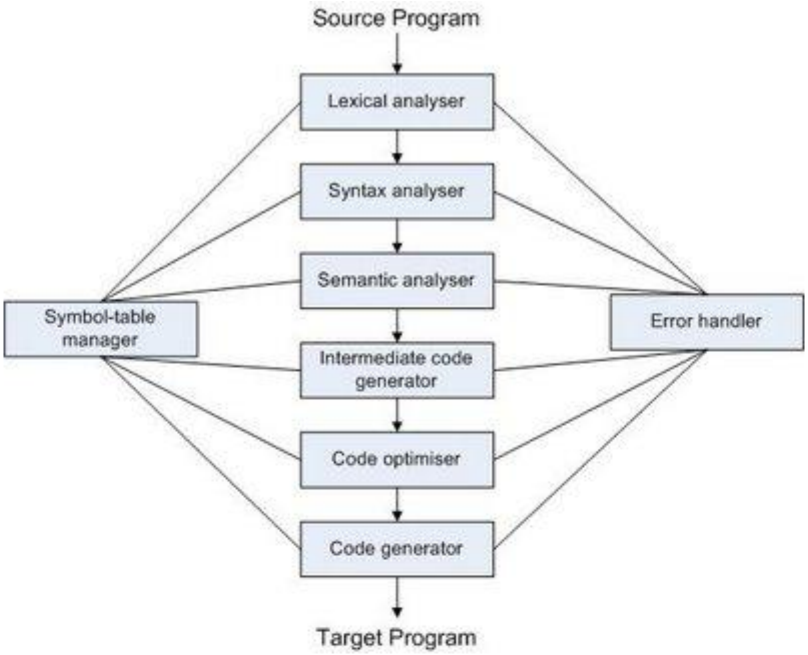
Heare also same... Some programmer (like me) writes a program with so many errors. If the compiler translates as it is, the computer may collapse (as it cant hit you for your errors)... So the compiler warns you about your errors so that you can correct them and resubmit your program. So, the main task of the compiler is to TRANSLATE, in-order to translate an Error free program, it warns you about your errors :)

The Technical Definition : So, technically you can say that "The compiler is a software program which converts the program written in High level language / Programming Language / User Language into the Machine Language / Binary Language to make an Executable Program.

The Phases of Compiler

There are several Phases in compilers are there. Those are

1. **Lexical Analyser** : Performs Lexical Analysis
2. **Syntax Analyser** : Performs Syntax Analysis
3. **Semantic Analyser** : Performs Semantic Analysis
4. **Intermediate Code Generator :** Performs Intermediate Code Generation
5. **Code Optimizer :** Performs Code Optimization
6. **Code Generator :** Generates Target Code



The **Symbol Table** is a Data Structure which is useful to store the identifiers in the program.

The **Error Handler** is useful in Detecting and Reporting the errors.

These two are useful from the starting to the ending of the conversion. (I mean from the starting phase to the ending phase).

NOTE : We can group these phases into two parts..

- **Analysis :** In which the compiler ANALYZES the given program. For this, the compiler SCANS and DIVIDES the program into parts. The first THREE phases comes under this category (**shortcut :** Check the names, if you can find the terms Analysis or Analiser, then that phase comes under the ANALYSIS category
- **Synthesis :** In which the compiler JOINS the parts of the divided program and makes the target program. (The last two phases comes under this category)

Now lets have a look at the functionalities of all these phases.

Lexical Analysis : It is nothing but analyzing the **LEXEMs (tokens),** in this phase, the compiler scans the entire program and pics the tokens and groups them.

Syntax Analysis : Syntax is nothing but the structure. It is nothing but to Analyze the structure of the program (whether it is correct or not)

Semantic Analysis : In this the compiler checks the MEANING of the program.

(Note : The above two process will execute simultaniously, so in some books they combine these two names and call the process as **Syntax and Semantic Analysis)**

Intermediate Code Generation : After scanning the program, the compiler generates an intermediate code (which is neither High level code nor Low level code, which looks like the Microprocessor Code)

Note : Most of the compilers generates the TAC (Three Address Code) as the Intermediate Code. It contains, maximum 3 variable, so we call it as the Three Address Code. (note : we can convert any problem into this TAC form).

it just look like, Temp1 = Temp2 + Temp3 (temporary variables)
      or Temp2 = Id + Temp4   etc,

Code Optimization : It helps to improve the Intermediate Code. It make assure that the target code will run EFFICIENTLY in LESS TIME (i mean, it should give the desired output in less amount of time)

Code Generation : It is the final phase. It generates the relocatable machine code or assembly code. It takes care about the target machine and where to store the variables in the target machine. Yea ofcourse, it takes care about the ORDER of the instruction execution too.

Operating Systems

1. What is an operating system?
An operating system is a program that acts as an intermediary between the user and the computer hardware. The purpose of an OS is to provide a convenient environment in which user can execute programs in a convenient and efficient manner. It is a resource allocator responsible for allocating system resources and a control program which controls the operation of the computer hardware.

2. Why paging is used?
Paging is solution to external fragmentation problem which is to permit the logical address space of a process to be noncontiguous, thus allowing a process to be allocating physical memory wherever the latter is available.

3. Explain the concept of the batched operating systems?
In batched operating system the users gives their jobs to the operator who sorts the programs according to their requirements and executes them. This is time consuming but makes the CPU busy all the time.

4. What is purpose of different operating systems?
The machine purpose workstation individual usability &resources utilization mainframe optimize utilization of hardware PC support complex games, business application Hand held PCs Easy interface & min. power consumption.

5. What is virtual memory?
Virtual memory is hardware technique where the system appears to have more memory that it actually does. This is done by time-sharing, the physical memory and storage parts of the memory one disk when they are not actively being used.

6. What is Throughput, Turnaround time, waiting time and Response time?
□ **Throughput** : number of processes that complete their execution per time unit.
□ **Turnaround time** : amount of time to execute a particular process.
□ **Waiting time** : amount of time a process has been waiting in the ready queue.
□ **Response time** : amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment).

7.What are the various components of a computer system?

□ The hardware
□ The operating system
□ The application programs
□ The users.

8. What is a Real-Time System?
A real time process is a process that must respond to the eventswithin a certain time period. A real time operating system is an operating system that can run real time processes successfully.

9.Explain the concept of the Distributed systems?
Distributed systems work in a network. They can share the network resources, communicate with each other.

10. What is SCSI?
SCSI - Small computer systems interface is a type of interface used for computer components such as hard drives, optical drives, scanners and tape drives. It is a competing technology to standard IDE (Integrated Drive Electronics).

11.What is a sector?
Smallest addressable portion of a disk.

12.What are the different operating systems?
□ Batched operating systems
□ Multi-programmed operating systems
□ Timesharing operating systems
□ Distributed operating systems
□ Real-time operating systems.

13.What is busy waiting?

The repeated execution of a loop of code while waiting for an event to occur is called busy waiting.

14.What are system calls?

System calls provide the interface between a process and the operating system. System calls for modern Microsoft windows platforms are part of the win32 API, which is available for all the compilers written for Microsoft windows.

15.What are various scheduling queues?

□ Job queue
□ Ready queue
□ Device queue

16.What are java threads?

Java is one of the small number of languages that support at the language level for the creation and management of threads. However, because threads are managed by the java virtual machine (JVM), not by a user-level library or kernel, it is difficult to classify Java threads as either user- or kernel-level.

17.What are types of threads?

□ User thread
□ Kernel thread

18.What is a semaphore?

It is a synchronization tool used to solve complex critical section problems. A semaphore is an integer variable that, apart from initialization, is accessed only through two standard atomic operations: Wait and Signal.

19.What is a deadlock?

Deadlock is a situation where a group of processes are all blocked and none of them can become unblocked until one of the other becomes unblocked. The simplest deadlock is two processes each of which is waiting for a message from the other.

20.What is cache memory?

Cache memory is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. As the microprocessor processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do the more time-consuming reading of data.

21. What is thrashing?

It is a phenomenon in virtual memory schemes when the processor spends most of its time swapping pages, rather than executing instructions. This is due to an inordinate number of page faults.

22. What are the states of a process?

□ New
□ Running
□ Waiting
□ Ready
□ Terminated

23.What is a binary semaphore?

A binary semaphore is one, which takes only 0 and 1 as values. They are used to implement mutual exclusion and synchronize concurrent processes.

24.What is a job queue?

When a process enters the system it is placed in the job queue.

25.What is a ready queue?

The processes that are residing in the main memory and are ready and waiting to execute are kept on a list called the ready queue.

26.What are turnaround time and response time?

Turnaround time is the interval between the submission of a job and its completion.
Response time is the interval between submission of a request, and the first response to that request.

27.What are the operating system components?

□ Process management
□ Main memory management
□ File management
□ I/O system management
□ Secondary storage management
□ Networking

□ Protection system

□ Command interpreter system

28.What is mutex?

Mutex is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously. When a program is started a mutex is created worth a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. the mutex is set to unlock when the data is no longer needed or the routine is finished.

29.What is Marshalling?

The process of packaging and sending interface method parameters across thread or process boundaries.

30.What are residence monitors?

Early operating systems were called residence monitors.

3**1.Why thread is called as a lightweight process?**

It is called light weight process to emphasize the fact that a thread is like a process but is more efficient and uses fewer resources(n hence “lighter”)and they also share the address space.

32.What are operating system services?

□ Program execution
□ I/O operations
□ File system manipulation
□ Communication
□ Error detection
□ Resource allocation
□ Accounting
□ Protection

33.What is a process?

A program in execution is called a process. Or it may also be called a unit of work. A process needs some system resources as CPU time, memory, files, and i/o devices to accomplish the task. Each process is represented in the operating system by a process control block or task control block (PCB).Processes are of two types

Operating system processes

- User processes

34.What are the different job scheduling in operating systems?

Scheduling is the activity of the deciding when process will receive the resources they request.

- **FCFS** FCSFS stands for First Come First Served. In FCFS the job that has been waiting the longest is served next.
- **Round Robin Scheduling** Round Robin scheduling is a scheduling method where each process gets a small quantity of time to run and then it is preempted and the next process gets to run. This is called time-sharing and gives the effect of all the processes running at the same time
- **Shortest Job First** The Shortest job First scheduling algorithm is a nonpreemptive scheduling algorithm that chooses the job that will execute the shortest amount of time.
- **Priority Scheduling** Priority scheduling is a scheduling method where at all times the highest priority process is assigned the resource.

35.What is dual-mode operation?

In order to protect the operating systems and the system programs from the malfunctioning programs the two mode operations were evolved

- System mode
- User mode.

36.What is a device queue?

A list of processes waiting for a particular I/O device is called device queue.

37.What are the different types of Real-Time Scheduling?

Hard real-time systems required to complete a critical task within a guaranteed amount of time.

Soft real-time computing requires that critical processes receive priority over less fortunate ones.

38.What is starvation ?

Starvation is a resourcemanagement problem where a process does not get the resources it needs for a long time because the resources are being allocated to other processes.

39.What is a long term scheduler & short term schedulers?

Long term schedulers are the job schedulers that select processes from the job queue and load them into memory for execution.

The Short term schedulers are the CPU schedulers that select a process form the ready queue and allocate the CPU to one of them.

40.What is fragmentation?

Fragmentation occurs in a dynamic memory allocation system when many of the free blocks are too small to satisfy any request.

41.What is context switching?

Transferring the control from one process to other process requires saving the state of the old process and loading the saved state for new process. This task is known as context switching.

42.What is relative path and absolute path?

□ **Absolute path -** Exact path from root directory.

□ **Relative path -** Relative to the current path.

43.What are the disadvantages of context switching?

Time taken for switching from one process to other is pure over head. Because the system does no useful work while switching. So one of the solutions is to go for threading when ever possible.

44.What is the state of the processor, when a process is waiting for some event to occur?

Waiting state

45.What is the difference between Primary storage and secondary storage?

□ **Main memory -** only large storage media that the CPU can access directly.

□ **Secondary storage -** extension of main memory that provides large nonvolatile storage capacity.

46.What is process synchronization?

A situation, where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called race condition. To guard against the race condition we need to ensure that only one process at a time can be manipulating the same data. The technique we use for this is called process synchronization.

47.What is a data register and address register?

 Data registers - can be assigned to a variety of functions by the programmer. They can be used with any machine instruction that performs operations on data.

 Address registers - contain main memory addresses of data and instructions or they contain a portion of the address that is used in the calculation of the complete addresses.

48.What are deadlock prevention techniques?

 Mutual exclusion

 Hold and wait

 No preemption

 Circular wait

49.What is the difference between Complier and Interpreter?

An interpreter reads one instruction at a time and carries out the actions implied by that instruction. It does not perform any translation. But a compiler translates the entire instructions.

50.What is a Safe State and what is its use in deadlock avoidance?

When a process requests an available resource, system must decide if immediate allocation leaves the system in a safe state. System is in safe state if there exists a safe sequence of all processes. Deadlock Avoidance : ensure that a system will never enter an unsafe state.

51.What is the difference between microkernel and macro kernel?

 Micro-Kernel : A micro-kernel is a minimal operating system that performs only the essential functions of an operating system. All other operating system functions are performed by system processes.

 Monolithic : A monolithic operating system is one where all operating system code is in a single executable image and all operating system code runs in system mode.

52.What is DRAM?

Dynamic Ram stores the data in the form of Capacitance, and Static RAM stores the data in Voltages.

53.What are the different functions of Scheduler?

Scheduler deals with the problem of deciding which of the process in the ready queue is to be allocated the CPU. Short Term Schedulers, Long Term Schedulers

54.What is a trap and trapdoor?

Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication. A trap is a software interrupt, usually the result of an error condition.

55.What are local and global page replacements?

Local replacement means that an incoming page is brought in only to the relevant process' address space. Global replacement policy allows any page frame from any process to be replaced. The latter is applicable to variable partitions model only.

56.What is cache-coherency?

In a multiprocessor system there exist several caches each may containing a copy of same variable A. Then a change in one cache should immediately be reflected in all other caches this process of maintaining the same value of a data in all the caches s called cache-coherency.

57.What are the benefits of multithreaded programming?

□ Responsiveness
□ Resources sharing
□ Economy
□ Utilization of multiprocessor architectures.

58. While running DOS on a PC, which command would be used to duplicate the entire

diskette? □ diskcopy

Software Engineering

Q) Some characteristics of software includes:-

1) Software is developed or engineer.
2) Most of software is custom build rather than assemble from existing component.
3) Computer program and associated documentation.
4) Easy to modified.
5) Easy to reproduce.
6) Software product may be developed for a particular customer or for the general market.

Q) Difference between program and software.

Program	Software
1) Small in size. 2) Authors himself is user-soul. 3) Single developer. 4) Adopt development. 5) Lack proper interface. 6) Large proper documentation.	1) Large in size. 2) Large number. 3) Team developer. 4) Systematic development. 5) Well define interface. 6) Well documented.

Definition of software: - it is systematic approach to the development, operation, maintenance and retirement of software. It is the application of computer science along with mathematics and ergative science. In the current scenario the S.E has a specific importance for making particular software**.**

Why software engineering:-

1) In the late 1960's hardware price were falling but software price rising.
2) Many software projects failed.
3) Large software project required large development loams.
4) Many software project late and over budget.
5) Complicity of software project is increased.
6) Demand for new software on the market.

Why study software engineering?

1) Higher productivity.
2) To acquire skills to develop large programs.
3) Ability to solve complex programming problems.
4) Learn techniques of specification design.
5) Better quality programmers.

Application of software:-

1) System software.
2) Application software.
3) Engineering/scientific software.
4) Embedded software.
5) Product line software.
6) Web application software.
7) Artificial intelligence software (AI).

1) The concept analysis in the view of S.E.
2) System development life cycle (SDLC).
3) Software requirement specification (SRS).
4) Object- data base and flow base analysis.
5) Models- spiral, water fall model.

1) The concept analysis in the view of S.E:- in the S.E the analysis phase helps to determine from the starting of the project to the end of the project. It has some specific phase. We can make particular software for real time use. The term analysis define the corresponding phase or stage by which the software developer can make a successful software. In the S.E the analyst has some specific job which is involved for making software. In a overview we can say that the software analysis is a main term through we can developed, operate and maintain a particular software. In the view of software developer as well as the user.

The software analysis we can determine by the view of two Side.

a) For the software developer.
b) For the customer (client).

System development life cycle (SDLC):- SDLC is an approach for making software for the developer, user and customer. SDLC focus on the internal phase to the end phase for making particular software. It generally deals with the analyst and the corresponding clients. SDLC has some specific phase. This are-

1) project identification
2) feasibility study
3) system analysis
4) system design
5) system development
6) system testing
7) system implementation
8) system maintenance
9) system documentation

1) **Project identification**: - in this phase the analyst focus the basic objective and identification need for the corresponding software. In this phase the analyst set up some meeting with the corresponding client for making the desired software.

2) **Feasibility study**: - feasibility defines in the three views for making particular software for the client. a) Technical b) financial c) social feasibility.

3) **System analysis**: **-** analysis defines how and what type of desired software we have to make for the client. It has some pen and paper base. Exercise through which the analyst focused for there desired goals.

4) **System design**: **-** in this phase the analyst draw the corresponding diagrams related to the particular software. in this phase the design include in the form of flow chat, data flow diagram, ntt relationship diagram (NRD).

5) **System development**: - development refers in the form of coding, error checking and debarking for the particular software. This phase deals with the developer activity for making a successfully software.

6) **System testing**: **-** testing refers whatever analyst and developer done will it be correct and error free to the desired software. In the S.E there some testing technique to which we can check whether project is error free.

Problem in the particular software. The main testing techniques are

1) white box testing
2) black box testing
3) ad hope testing
4) system testing
5) unit testing
6) alpha testing
7) beta testing

7) **System implementation**: - after completing the testing phase we have to implement a particular product or system according to the customer need. In the implementation phase some design and other user activity part may be changed as per customer need.

8) **System maintenance**: - after implementation the users use the particular software to there corresponding operation to active there job. In this phase the software maintained from the user or developer side after spanning some times of use of particular software. In this phase the related hardware, software and other utilities are also maintained.

9) **System documentation**: - documentation refers the approach and guidelines for the user as well as the customer to the related software. The documentation refers some writing instruction for how to use for related hardware requirement, and also some maintains factors for the users.

Srs (software requirement specification)-

Definition- SRS is a complete reading base documentation focus on the particular desired software to the specific client or customer. After collecting the necessary data from SDLC we have to summarize the useful and appropriate data for making desired software. SRS has some objectives which is help to the software developer as well as the customer for making a successfully software.

Characteristics of SRS-

1) complete
2) traceable
3) appropriate for the developer
4) modifiable
5) simple language
6) software requirement view

Good SRS: - SRS is a very important for fact gathering technique which include the consider delimiting with the customer gather previous information related to particular software and returned on investment the good characteristics of SRS includes the following activities:

1) It focuses on summarized from for a particular software specification.
2) The completeness deserves the related phase for making a required specification.
3) The traceable factor focus on the modification part when ever it necessary for the development of SRS.

Waterfall model: - waterfall model linear representation of its face for developing a particular system in this model. These phases are:

1) Communication (requirement analysis, specification).
2) Planning (estimating, seducing).
3) Modeling (analysis, design).
4) Construction (code, test).
5) Deployment (delegacy, support, fid back).
   a) This phase involve the whole working process for making a particular software for desired client.
   b) Iterative waterfall model and the classical waterfall model are the two types are these model available calices. We have to follow the linear stage or phase which are available in waterfall model. But in the iterative model we can jump over from one phase to another phase.

Disadvantages of waterfall model:-

1) In real life the project are sequential.
2) Intel finished the project the working versa is not available for the customer.

3) Hardware implementation in the middle of the project is fact.
4) In this model the risk factor is not available.

Spiral model: - the spiral model is a step by step process it stresses on the risk factor of the SDLC.

The stages of the model are:

1) planning
2) risk analysis
3) development
4) customer assessment
    a) determine objective alternative
    b) evaluative alternative risk analysis
    c) development and product
    d) plane for next phase

From the above figure we can see that there are four parts including some loops how ever the number of loops is not fixed depending on the project. Spiral model has a special type of focus on risk factor by which it has make some difference from the other model. In this model the phases are planning, risk analysis, development and customer assessment. From the diagram we can say that first quadrant which representation and make some object and determine the analysis for a particular project. The next part deals and evaluated some alternative way by which we can achieve a simple solution. In this phase the risk analysis is also measured depending on the project.

Difference between system analysis and system design-

System analysis	System design
1) System analysis is the examination of the problem. 2) It is concerned with identifying all constrains. 3) It deals with data collection and a datelined evaluation of existing file. 4) In system analysis part the main focus on data flow diagram and data dictionary.	1) System design is the creator of information system which is solution to the program. 2) It is concerned with co erudition of the activities for a particular system goal. 3) It deals with general design specification detailed design specification I/O files and procedures. It also deals with program construction testing and user acceptance. 4) It provides technical specification and reports by which problems can be trapped.

Difference between program and engineering.

Program	Engineering
1) Small project. 2) You 3) Once product. 4) Cheap 5) Few sequential changes.	1) large product 2) team 3) family of product 4) Costly 5) Many parallel changes.

Module 4

System planning

1) What is system planning?

By the planning in S.E it refers the whole internal and external working activities for making particular software. the planning involves the software developer and client base future planning from the starting to the ending phase for making a successful software.

The planning depend upon some categorize by which the software development team.

2) Data and fact gathering.

For making a particular planning the data and fact are most to important factors for set up a good plan for the development particular software.

For collecting data and fact from the real world. Some techniques are interviewing, site visit, previous software data gathering and some real world based examples software to the particular clients.

This technique are very essential and important due to original, meaningful and exact right information through which the developer (software project team) for set up there plan for making particular software.

3) **planning in the view of S.E: -**

In the view of S.E planning is very essential in the view of creating implementing, error checking, cost estimation schedule and maintenance for a particular software. in this part the development team and the project team set a focus for making a particular software for the desired client. The planning has some advantage for the software developer as well as the user for the particular software.

These are:

1) interviewing
2) communication
3) presentation
4) site visit

1) **Interviewing**: - it is the method by which we collect the data by a specific communication by the software developer and the customer. It is a face to face approach by which we can assume that what actually customer wants.

2) **Communication**: - it refers a technique for building a project at the start phase. Communication here means dialog between two authorities. i.e., software development team and the customer from the software company view generally we make a team for the communication purpose.

3) **Presentation**: - it can be made from the both side. i.e., software development team and customer. It is basically a technique whose we can understand that what actual they deal.

4**) Site visit**: - it is a very important technique for collecting data for a specific operation. In the site visit a development team visits the customer site and tries to understand that what actual operation they do.

Module 4

System design

1) Discuss the term of system design: -

After getting all information regarding making and implementation of software from system development life cycle or from any source which helps to build successful software. Design part in view of software is very important relating to the other issues in the software development process. Design here means that the structure and all essential steps, which is to be used for making a particular software. These are implemented with a specific diagram.

In S.E the design part generally controlled the system analysis parts which are maintained by the system analyst. In view of S.E and information SAD, it has two types of specific design-

a) DFD (data flow diagram).

b) ERD (entity relationship diagram)

2) **What do you understand by design** (in view of S.E): - a model or design is an abstract representation of a set of process? Each model represents a process and data from a particular perspective model provide person information about some data and process. The genetic models do not contain process description rather it deals with product. Different processes are used to develop different part of the model.

System design tools are basically in two types.

a) System modifying tools (DFD, ERD).

b) System design tools (data dictionary, process description). The structured analysis involves-

a) Data flow diagram (DFD).

b) Data dictionary.

c) Process description.

In 1976 chain, introduced the method of data analysis in which takes please through out whole development cycle but in different degrees in detail. Usually analysis starts early by developing to level conceptual modeling with using some specific symbols from a modeling method known as entity relationship analysis. It starts when a DFD is finished.

DFD (data flow diagram) - the function or process and the data items that are exchanged between different function are represented in a diagram known as DFD. The program structure is designed from its DFD representation. A DFD shows the logical flow of data through a transaction processing system with out regard to time period when each function occurred. They are used in system development process. Symbol used in DFD:

Lists □ external data source or data destination.

o- Internal entity or transformation process.

Context diagram: - DFD is a graphical representation of data flow. Source process and output are drowning with particular symbols. How ever the diagram close not contains any process logic or any conditional flow. When a single process connects all data source with proper arrows. The DFD is called context or zero level DFD. A context diagram is decomposed into successfully lower level in detours. These types of DFD are referred as level 1, level 2…

Physical DFD: **-** the first level of details in a DFD in generally called a physical DFD. It focuses on physical entities involved in the system under study as well as the readable documents reports and other hard copy of input and output.

Logical DFD: - it deals with participant to each bottle contain. That indicates a task the system performs. Logical DFD helps the designers to decide what system resources are available, what acuities how to protected control this system after instruction. Context level, DFD for payroll processing system.

Time card data

pay checks

Time Keeping

Employee

Payroll Processing system

Payroll summers

Payroll change Data

Management

Human Source

government agencies

ERD: -

Definition: - an ERD is a graphical model of information system that describes the relationship among the system entities. ERD is a major data in our project into entities and define the relationship between the entities.

Component of ERD: - an ERD has 4 major graphical components:

1st entity: - a data entity is anything read or abstract about which we want to solve and stored in a particular project for example employee's books, computers etc.

2nd entity: - a data relationship is a natural association that exists between one or more entities.

3rd attribute: - an attribute is a characteristic of any particular entity which we can explain in our practical file, for example if we consider the entity is an employee, then the attribute are emp_name, emp_address emp_id designation etc.

4th cardinality: - it is the specific no of occurrence from one object to another object or one entity to another entity for a particular description.

Data dictionary: - data flow analysis (DFD) consists of a 4 tools. These are:

1) data flow diagram
2) data dictionary
3) data structure diagram
4) structure chart

Data dictionary contain the information about the data of a system. I.e. the data about data or Meta data.

A data dictionary is organized into 5 sections:

a) Data elements
b) Data flows
c) Data stores
d) Process
e) External elements

Object oriented design (OOD): - in the object oriented design the analyst consider and implement the entities in a particular object. In the other design tools, the global data base concept has implemented for a particular project but in the case of OOD the main focus of haring the data through a massage passing through one to another.

In the real world scenario this type of design tools has made its own success. Due to dealing with the object and the object oriented concept in the specific design method. In the S.E the object oriented design has some specific approach as a design tool for analyst.

Coupling and cohesion: -

Coupling: - between two methods indicate the degree of independence between them. The degree of coupling between two modules depends on their interface complicity. It is basically determined by no of parameters that are interchange while invoking function of a particular module. Depending on their interface complicity; following five coupling can occur between two modules

Data	stamp	control	Common	content

Low → high

Classifications of coupling: -

Data coupling: - two modules are data coupled if they communicate using same data item that is passed as a parameter between two modules. For example- an integer, a float etc.

Stamp coupling: - two modules are stamp couple if they communicate using a composite data item. Such as a structure in c, as a record Pascal.

Control coupling: - this exists between to modules of data from one module is used to direct which helps to execute for another module.

Cohesion- it is a measures of the functional strength of a particular module. A module having high cohesion and low coupling is said to be functionally, independent to other module. By term functional independence we mean that a cohesion module performs a single task for a single function. We use cohesion due to following lesion

1) error ideation
2) scope of reuse
3) understandability

Classification of cohesion-

Coinaolental
Logic
temporal
Procedural
Communicational
Sequential
function

Module 5

System documentation

In s/w we can the basic two factors these are 4p's concept and 4w's concept

4p's- people process, product and plan
4w's- who, where, what, when.

Documentation is a serial part in the S.E method of for making its own success for internal and external users. Documentation refers the whole process relating to the operation, use component Tec knowledge and other information relating to particular software.

Types of documentation: - in broad of view the documentation can be classified in two types-

a) internal documentation
b) external documentation

In external documentation it can be again directed in 7 parts:

1) management documentation
2) system documentation
3) operational documentation
4) user documentation
5) program documentation
6) training documentation
7) implementation documentation

Internal documentation refers the internal work process like coding, programming, structure which helps software analyst and programmer.

The external documentation deals with the user manual and other related information for particular plan (customer).

1) **User documentation**: - it refers the whole user details for particular software and something it refers how the are going on in a particular software.

2) **Programming**: - these types of documentation help the programmer and other system level manager to directly or indirectly involve with programming structure.

3) **System**: - the goals of to focus on the software as well as hardware component in which particular software run. It has some specific features like system overview, SRS, design, test and implement.

4) **Management**: - it is used for the upper level manager and its contains the whole work project plan including time and cost for a software.

5) **Operational**: - it maintains the internal working process including some problem solution for the customer. It is a basically a guest of how to use the software to the client.

6) **Training**: - it basically is a summarized from for training purpose and user guideline of a software. it helps for the training purpose people who wants to use a particular software in their specific application.

7) **Implementation**: - it generally helps to internal process. It describes how the software will be implementing in a particular system. This document is very important for software developer and customer of how the software implements and run for a specific purpose.

System documentation: - it has the following features for its own its:

1) system overview
2) SRS
3) Specification/design/implementation
4) Test plan
5) Data dictionary
6) Acceptance of test plan.

1) **System overview**: - it is very important step for making particular software for the software developer. System overview is a part of the documentation for use specific software. it has some software as well as some hardware information which is used for the software developer.

2) **SRS**: - it is important features for the system documentation. It has some specific goals which help for making particular software.

3) **Specification/design/implementation**: - the part deals with the system analyst that describe how to use, how to implement, how to make and how to make maintain the particular software including design phase.

4) **Test plan**: - it refers the hole works must be tested with some specific steps and parameters for deciding correctness for a particular work.

5) **Data dictionary**: - it contain data about data (Meta data) this data's are useful for implementation from the coder side.

6) **Acceptance of test plan**: - this feature include in the system documentation part to know whether the design implementation and progress is accepted by designer developer and some time for user.

Principles of system documentation: - it has the following specific features. This are

1) availability
2) objectivity
3) cross referencing
4) easy to maintain
5) completeness

1) **Availability**: - it refers the documentation should be available for the analyst, developer and the user. In right please at right time.

2) **Objectivity**: - the objectivity must be focused and clear who are making the documentation for its specific use. Objectivity refers the original method for making particular software.

3) **Cross referencing**: - it determines for internal communication between to or more module. This feature includes the internal relational ship between the modules.

4) **Easy to maintain**: - after complete the documentation some times it refers in future it may be changed at this time the documentation should be worked properly and correctly.

5**) Completeness**: - it refers the all phase including design, coding, testing, user manual. This must be included for making the documentation successful and complete.

Module 6

Coding and programming

The main object of coding is to implement the design with the help of some programming environment to achieve a specific task. After writing the code we have to go through the error checking part to run the code successfully. After that the code and the design part we have to match those we the original output we may get from the original code.

Coding techniques helps to increase the cast of the software in the error handing part and the implementation part.

Choice of programming language: - to achieve the particular software design output in the real world the coder has to choice. Some programming language with some specific programming environment for this we have to know very well the following objectives in the software coding part

1) Understand very well the design approach.
2) Enough knowledge of programming and environment.
3) Select a programming language to achieve a specific task for client.
4) Understand the original cost and coding part in the real world scenario.

Basically we have two types environment available in software industry

a) structured programming
b) object programming

The both programming environment has some specific norms in its own area. For example the structured programming environment deals with some specific function with its own parameter an OOP environment based on the objectives which is sharable from method to another.

Mixed language programming: - it is basically a integrated approach for the software coder to direct and indirect they can change the environment as it required at the time of operation.

The environment helps the customization for a specific software design and it is very under friendly environment for user. For example ERD these type of software like SAP Microsoft lavation. This type of software as some mixed programming language environment. Which helps the coder as well as the user?

Coding structure: - whenever the programmer writes some codes from the design view. He/she has peach a programming environment achieved the desired goal. Depending programming language and environment we have to follow norms and syntax of that programming environment coding principle are –

1) construct the algorithm
2) select the data structure that will meet the need of design,
3) understand software architecture and create a specific interface,
4) keep conditional knowledge as simple as possible,
5) create nested loop that marks their easily testable,
6) select meaningful variable and follow other local coding instruction,
7) write code that is self documenting,
8) Create a visual laired for easy understanding.

Module- 7

System testing

Definition of testing: - according to Myers (1979) testing is technique of software to find the error or mistake in a particular software design.

The term over we can define as some unconditional or conditional, direct or indirect mistakes from our design phase to implementing phase in particular software area. In testing the main fault and failure are two main corresponding factors in testing. Fault is a condition for required achievement of a specific function where as failure is the inability to achieve a desired goal.

Software testing is a very essential part in S/E to make a product success. Its basic objective for checking all parts connecting through a specific project.

Types of testing: -

Testing

Unit

1) ad hope
2) black box
2) white box

integration

1) top down
2) bottom up
3) regression
4) Smoke

System

performance, stress, volume, configuration, regression, recovery, maintenance, alpha-testing, beta-testing, compatibility, And documentation.

acceptance

Unit testing: - in testing methodology first testing is unit testing. It has basically three parts a hope testing, white box testing, black box testing.

Ad hope testing: - in ad hope testing we can test whatever comes in mind for a specific testing. It is sometimes called random testing process. In this technique the scope and limitation of software is absent.

White box testing: - in white box testing the structure of the code. We can test without logic. In code segment white box testing are basically in 7 types: -

1) segment testing (each segment to control structure)
2) branch coverage or node testing (each blank)
3) basis path testing (predetermined path setting)
4) compound condition coverage (or, and)
5) data flow testing (variable data flow)
6) path testing (all paths are covered or defined)
7) Loop testing.

Black box testing: - this type of testing for unknown logic structure – in this the function oriented modules are tested; it includes the following activities -

1) error crossing
2) equivalence class classification
3) boundary value analysis
4) cause and effect graphic
5) domain testing
6) module interface testing
7) Command line testing.

Integration testing: - in the integration testing when we use more than one environment to perform software then use integration testing. In this it has basically 4 parts

1) top down testing
2) bottom up testing
3) regression testing
4) smoke testing

1) **Top down testing**: - in this test each module or class from the top level of program segment and after that we can add the first one to next and continue this process until complete module check.

2) **Bottom up testing**: - in this testing it is a reserve case of to down testing but a simple difference is that this does not allow to integration the tested modules from bottom up.
3) **Regression testing**: - this type of testing define and examine with some existing function and future achievement goal in specific software regression testing allows the software development team to ensure that the existing functions and other modules whether they can use for their coming project or not.
4) **Smoke testing**: - it defines the testing approach where we can implement and test each and every module and function with the existing project. This type of testing approach is basically needed. When we need to build up a long term project. Inside of project if software development team wants to test some part of code or module they may up through smoke testing.

System testing: -

1) **Performance testing**: - carried out to check weather system meets the non functional requirements in SRS documents.
2) **Stress testing**: - this type of testing help to evaluate the system performance when it is stressed for a sort period of time.
3) **Volume testing**: - this type of testing helps to check weather data structure, stack, array queue etc. has been classing successfully for a particular solution.
4) **Configuration**: - use to analysis various types of system behavior including hardware and software for a specific requirement.
5) **Compatibility**: - this type of testing is required when system interface with other type of system. This helps to check weather interface function performs its corresponding task or not.
6) **Regression**: - it is necessary when system is being tested when a system may be updated from an existing system to fix some bugs or error in that particular environment.
7) **Recovery**: - this helps to determine of the software or system will be failure for any résumé, then there should be a particular way to recover the data and information from previous one.

8) **Maintenance**: - this test the all necessary states from design to implement and determine weather in future all types of altercation will be available.
9) **Documentation**: - this check ensures that all types of required modules over technical manuals are existing in that particular software project.
10) **Alpha testing**: - it is system testing performed by development team within the developing organization.
11) **Beta testing**: - it is system testing performed by selected group of customer.
12) **Acceptance testing**: - this type of testing is done by the users to determine that software will be accepted or rejected.

Validation and verification: - validation refers to the set of authorities which ensure that software is correctly implemented for a specific function. Verification refers to a different set of authorities which ensure that software which has been built is traceable to customer requirements.

Verification: - are we making the product?
Validation: - are we making the right product.

It is basically a part of the software quantity assurance (SQA): - tested approach in the S/E-

It includes the following objectivity-

1) To ensure that product is sofa and result.
2) To ensure that product will function properly under both normal exceptional conditions.
3) To ensure that the product is what the user want.

Module 9

Software quality

Explain quality: - achieving a high level of product or service, the quality is the objective of this most organization. It is longer acceptable to the customer of a particular poor quality product and then it refers after they have been delivered to customer.

Software quality management can be structured in three activities: -

Quality assure: - the established next of a frame work of organizational procedures and standard which lead to high quality software.

Quality planning: - the selection appropriate procedures and standard form this frame work and adaptation for a specific software project.

Quality control: - the definition of a process which ensures that project quality procedures and standards are followed by software development team.

Quality tools: - an international standard team can be used in development of a quality management system In all industries is called ISO (international standard organization). ISO-9000 is a set of standards that can be applied to a range of organization from manufacturing to service industries ISO-9000 three interrupts ISO-9000 for software development. These tools include –

1) design control
2) product identification and traceability
3) inspection and testing
4) corrective action
5) document control
6) Service training etc.

Quality assurance: - it consist of a set of checking and reporting that perform with quality control activities of includes: -

1) correctness
2) integrity
3) maintains
4) Check for particular client.

Quality assurance activities defines frame work for achieving software quality which involves with selecting standards the should be applied for software development process or software product. The product and process standards are two vital factors for software quality assurance.

McCall's quality factors: - the factors that effect software quality can be categorized into two groups –

1) Factors that can be directly measured.
2) Factors that any measured indirectly (usability).

In each case segment the focus is to defect the quality factor in the software including program data and documents.

McCall's Richards and want user proposed a useful structure of quality factor that helps the software quality. The main three objectives are that: -

1) Operational characteristics
2) Ability to under go change.
3) Adaptability to new document.

McCall's quality factor has 3 main parts-

a) product revision
   1) maintainability
   2) flexibility
   3) testability
b) product transition
   1) portability
   2) reusability
   3) interoperation ability
c) product operation
   1) correctness
   2) integrity
   3) effectiveness
   4) reliability
   5) usability

Module 10

Software project management

Explain project management: - project management is the technique of maximizing the provability that a project developer its goal on time, within the budget and the good quality. Here the 4 P'S connect-people, process, product and plan are key factors for the whole project management.

A common process frame work is selected an appropriate S/E. paradigm through is applied and a set of work is done to a specific project management are –

1) Project planning
2) Project scheduling
3) Risk management
4) Managing people
5) Software cost estimation

Software project management has some specific goal to make it successful on the particular software making after 1993 the software quality management and risk management were involved with software project management.

Project control: - in software industry the main problem of making a software are –

1) estimate the cost
2) knowledgeable professional
3) delivery within time
4) meets all customer requirement
5) good quality
6) Scope for the future.

To solve this problem the software project management helps to achieve for particular software run with its own success and customer satisfaction. We know the main elements of software project management are –

Project planning: - it has some specific objectives. These are – a) determine the requirement

b) determine the resource
c) select a particular life cycle
d) determine project strategies

Project planning is very essential from starting phase to ending phase of software.

Project scheduling: - it refers the time span or period for a specific software project. It is basically 3 components:

a) plan
b) effectiveness
c) scheduling

With help of some charts and other graphical tools helps to determine the schedule of a particular project. Here resource allocation the professional people and other related categorized are involved with project scheduling.

Risk management: - in software industry the risk term is often use the relation of a failure and successful project. In real world scenario sometimes the resource are not available as much we want for that point of view the risk term played a specific job for software developer as well as the customer.

Managing people: - it represents the professional knowledgeable person who deals a particular project as a managerial and technical view.

Software cost estimation: - it is very important for software project management to control the overall budget from company side as well as customer or client SID. Cost estimation in S/E has some specific norms and rules to estimate of particular software.

Quality management: - quality is another key factor for particular software for a development team as well as customer ideology for any industry. ISO-9000-3 look after the original quality of software in software market.

WBS (work break down structure): - a WBS in project management and system engineering is use to define and group a project. Work elements in a way that helps organization and define that total work scope of the project.

A WBS element may product, data service or any combination. A WBS also provide the necessary from was for detailed cost estimating and control along with providing guidelines for schedule development control. Additionally the key is a dynamic tool that can be developed and received as needed by project manager. A WBS is structure which shows sol division of effort required to achieve an objective.

For example- a program a project and contract the project or contract the WBS is developed by starting with the end objectives and solve dividing into manageable components in tare-of-size, duration and responsibility.

One of the most important WBS, structure design principle is called 100% rule. The 100% rule state that WBS includes 100% of the work defined by the project scope and capture for internal and external process in terms of work to be completed including Project management.