NAAC ACCREDITED

तेजस्वि नावधीतमस्तु

ISO 9001:2008 & 14001:2004

FAIRFIELD

Institute of Management & Technology

'A' Grade Institute by DHE, Govt. of NCT Delhi and Approved by the Bar Council of India and NCTE

Reference Material for Three Years

Bachelor of Computer Application

Code : 020

Semester – IV

FIMT Campus, Kapashera, New Delhi-110037, Phones : 011-25063208/09/10/11, 25066256/ 57/58/59/60
Fax : 011-250 63212 Mob. : 09312352942, 09811568155 E-mail : fimtoffice@gmail.com Website : www.fimt-ggsipu.org

INDEX

Three Years

Bachelor of Computer Application

Code : 020

Semester – IV

1) Descrie the concept of permutation & combination.

ANS:= Permutation and combination has lately emerged as an important topic for many entrance examinations. This is primary because questions from the topic require analytical skill and a logical bend of mind. Even students who do not have mathematics as a subject can handle them if they have a fairly good understanding of the concepts and their application. Hence anyone who is well-versed in different methods of counting and basic calculations will be able to solve these problems easily

IMPORTANT NOTATION

$n!$ (Read as n factorial)

Product of first n positive integers is called n factorial

$n! = 1 \times 2 \times 3 \times 4 \times 5 \times \ldots n$

$n! = (n - 1)!\ n \in N$

In special case $0! = 1$

Permutation

The arrangement made by taking some or all elements out of a number of things is called a permutation.

The number of permutations of n things taking r at a time is denoted by ${}^nP_r$ and it is defined as under:

$${}^n p_r = \lfloor n / \lfloor (n-r) \rfloor$$

Combination

The group or selection made by taking some or all elements out of a number of things is called a combination.

The number of combinations of n things taking r at a time is denoted by ${}^nC_r$ or and it is defined as under:

$${}^nC_r = \lfloor n / \lfloor r \lfloor\ (n - r)$$

Here $n!$ = Multiple of n natural number

Some Important Results of Permutations

1. ${}^nP_{n-1} = {}^nP_n$
2. ${}^nP_n = n!$
3. ${}^nP_r = n\,({}^{n-1}P_{r-1})$
4. ${}^nP_r = (n - r + 1) \times {}^nP_{r-1}$
5. ${}^nP_r = {}^{n-1}P_r + r\,({}^{n-1}P_{r-1})$

Types of Permutations

When in a permutation of n things taken r at a time, a particular thing always occurs, then the required number of permutations = $r\,({}^{n-1}P_{r-1})$.

EXAMPLE:- If ${}^nC_{10} = {}^nC_{14}$ then find the value of n

Solution

${}^nC_{10} = {}^nC_{14} \Rightarrow n = (10 + 14) = 24 \quad (\because n = p + q)$

Permutations with Repetition

These are the easiest to calculate.

When you have ***n*** things to choose from ... you have ***n*** choices each time!

When choosing ***r*** of them, the permutations are:

n × n × ... *(r times)*

(In other words, there are **n** possibilities for the first choice, AND THEN there are **n** possibilities for the second choice, and so on, multiplying each time.)

Which is easier to write down using an exponent of **r**?

n × n × ... (r times) = n^r

Example: in the lock above, there are 10 numbers to choose from (0,1,..9) and you choose 3 of them:

10 × 10 × ... (3 times) = 10^3 = 1,000 permutations

2) Explain Binomial Distribution with the help of example.

ANS:- BINOMIAL THEOREM

For any positive integral value

$(x + a) = nc_0x^n + nc_xx^{n-1}a + nc_2x^n - {}^2a^2 + \ldots + {}_na^n$

Proof

We can prove this theorem by using principle of mathematical Induction. First we shall verify the theorem for $n = 1$

∴ The result is true for some positive integral value of k or n.

$\therefore (x + a)^k = kc_0x^k + kc_1x^{k-1}a + kc_2x^{k-2}a^2 + \ldots + kc_ka^k$

Now multiplying both sides by x and a and adding the two results

$x\,(x + a)^k = kc_0x^{k+1} + kc_1xa + kc_2x^{k-1}a^2 + \ldots + kc_k\,xa^k$

and

$a(x + a)^k = kc_0x^ka + kc_1x^{k-1}a^2 + kc_2x^{k-2}a^3 + \ldots + kc_kxa^{k+1}$

$\therefore (x + a)^{k+1} = kc_0x^{k+1} + (kc_1 + kc_0)x^ka + (kc_2 + kc_1)x^{k-1}a^2 +$

$(kc_3 + kc_2)x^{k-2}a^3 + \ldots + kc_ka^{k+1}$

By using ${}^nc_r + {}^nc_{r-1} = {}^{n+1}c_r$ we can say that

$(x + a)^{k+1} = {}^kc_0x^{k+1} + {}^{(k+1)}c_1x^ka + {}^{(k+2)}c_2x^{k-1}a^2 + \ldots + {}^kc_ka^{k+1}$ *

GENERAL TERM OF BINOMIAL EXPANSION

We have derived binomial expansion as

$(x + a)^n = {}^nc_0x^n + {}^ac_1x^{n-1}a + {}^nc_2x^{n-2}a^2 + \ldots {}^nc_na^n$

Here $T_1 = {}^nc_0x^n$

$T_2 = {}^nc_2x^{n-1}a$

$T_3 = {}^nc_2x^{n-2}a^2$

∴ The $(r + 1)$ th term of the binomial expansion can be written as follows: $T_{r+1} = {}^nc_rx^{n-r}a^r$

The (r + 1) th term of binomial expansion is called the general term of expansion.

3) Explain probability & conditional probability.

ANS:- Probability:-

Probability is the likely percentage of times an event is *expected* to occur if the experiment is repeated for a large number of trials. The probability of rare event is close to zero percent and that of common event is close to 100%. Contrary to popular belief, it is not intended to accurately describe a single event, although people may often use it as such. For example, we all know that the probability of seeing the head side of a coin, if you were to randomly flip it, is 50%. However, many people misinterpret this as 1 in 2 times, 2 in 4 times, 5 in 10 times, etc

$$P(A|B) = \frac{P(A,B)}{P(B)}.$$

Conditional Probability

$P(A|B)$

Conditional probability is the probability of one event occuring, given that another event occurs. The following expression describes the conditional probability of event A given that event B has occurred:

If the events A and B are dependent events, then the following expression can be used to describe the conditional probability of the events:

$$P(B \mid A) = \frac{P(A,B)}{P(A)}.$$

$$P(A,B) = P(B|A) * P(A) = P(A|B) * P(B)$$

This can be rearranged to give their joint probability relationship: This states that the probability of events A and B occurring is equal to the probability of B occurring given that A has occurredmultiplied by the probability that A has occurred. A graphical representation of conditional probability is shown below:

4) Discuss Baye's Theroem.

Ans:- Bayes' Theorem

Most probability problems are not presented with the probability of an event "A," it is most often helpful to condition on an event A"." At other times, if we are given a desired outcome of an event, and we have several paths to reach that desired outcome, Baye's Theorem will demonstrate the different probabilities of the pathes reaching the desired outcome. Knowing each probability to reach the desired outcome allows us to pick the best path to follow. Thus, Baye's Theorem is most useful in a scenario of which when given a desired outcome, we can condition on the outcome to give us the separate probabilities of each condition that lead to the desired outcome.

Derivation of Baye's Theorem: - The derivation of Baye's theorem is done using the third law of probability theory and the law of total probability.

Suppose there exists a series of events: $B_1, B_2, \ldots, B_n$ and they are mutually exclusive; that is,

This means that only one event, B_j, can occur. Taking an event "A" from the same sample space as the series of B_i, we have:

Using the fact that the

events AB_i are mutu

$$P(A) = \sum P(A \mid B_j)P(B_j)$$

5) Describe Probability Distribution.

Ans:-- PROBABILITY DISTRIBUTIONS

In any probabilistic situation each strategy (course of action) may lead to a number of different possible outcomes. For example, a product whose sale is estimated around 100 units, may be equal to 100, less, or more. Here the sale (i.e., an outcome) of the product is measured in real numbers but the volume of the sales is uncertain. The volume of sale which is an uncertain quantity and whose definite value is determined by chance is termed as *random (chance* or *stochastic) variable.* A listing of all the possible outcomes of a random variable with each outcome's associated probability of occurrence is called *probability distribution*. The numerical value of a random variable depends upon the outcome of an experiment and may be different for different trials of the same experiment. The set of all such values so obtained is called the *range space* of the random variable.

6) WHAT ARE EXPECTED VALUE AND VARIANCE OF A RANDOM VARIABLE .

ANS:- **Expected Value** The mean (also referred as **expected value) of a random variable** is a typical value used to summarize a probability distribution. It is the weighted average, where the possible values of random variable are weighted by the corresponding probabilities of occurrence. If x is a random variable with possible values $x_1, x_2,\ldots, x_n$ occurring with probabilities $P(x_1), P(x_2),\ldots, P(x_n)$, then the expected value of x denoted by $E(x)$ or μ is the sum of the values of the random variable weighted by the probability that the random variable takes on that value.

7) Discuss Binomial Probability Distribution

Ans:- Binomial probability distribution is a widely used probability distribution for a discrete random variable.

This distribution describes discrete data resulting from an experiment called a *Bernoulli process* (named after Jacob Bernoulli, 1654-1705, the first of the Bernoulli family of Swiss mathematicians). For each trial of an experiment, *there are only two possible complementary* (*mutually exclusive*) outcomes such as, defective or good, head or tail, zero or one, boy or girl. In such cases the outcome of interest is referred to as a '*success*' and the other as a '*failure*'. The term 'binomial' literally means two names.

Bernoulli process: It is a process wherein an experiment is performed repeatedly, yielding either a success or a failure in each trial and where there is absolutely no pattern in the occurrence of successes and failures. That is, the occurrence of a success or a failure in a particular trial does not affect, and is not affected by, the outcomes in any previous or subsequent trials. The trials are independent.

8) What is Poisson Probability Distribution

Ans:-Poisson distribution is named after the French mathematician S. Poisson (1781-1840), The Poisson

process measures the number of occurrences of a particular outcome of a discrete random variable in a *predetermined time interval, space, or volume*, for which an *average number* of occurrences of the outcome is known or can be determined. In the Poisson process, the random variable values need counting. Such a count might be (i) number of telephone calls per hour coming into the switchboard, (ii) number of fatal traffic accidents per week in a city/state, (iii) number of patients arriving at a health centre every hour, (iv) number of organisms per unit volume of some fluid, (v) number of cars waiting for service in a workshop, (vi) number of flaws per unit length of some wire, and so on. The Poisson probability distribution provides a simple, easy-to compute and accurate approximation to a binomial distribution when the probability of success, p is very small and n is large, so that $\mu = np$ is small, preferably $np > 7$. It is often called the *'law of improbable'* events meaning that the probability, p, of a particular event's happening is very small. As mentioned above **Poisson distribution** occurs in business situations in which there are a few successes against a large number of failures or vice-versa (i.e. few successes in an interval) and has single independent events that are mutually exclusive. Because of this, the probability of success, p is very small in relation to the number of trials n, so we consider only the probability of success.

9) Explain about Variance and Standard Deviation

Another way to disregard the signs of negative deviations from mean is to square them. Instead of computing the absolute value of each deviation from mean, we square the deviations from mean. Then the sum of all such squared deviations is divided by the number of observations in the data set. This value is a measure called **population variance** and is denoted by $\boldsymbol{\sigma}^2$ (a lower-case Greek letter sigma). It is usually referred to as 'sigma squared'. Symbolically, it is written as:

Population variance,

where $d = x$ - A and A is any constant (also called assumed A.M.)

Since $\boldsymbol{\sigma}^2$ is the average or mean of squared deviations from arithmetic mean, it is also called the *mean square average.*

The population variance is basically used to measure variation among the values of observations in a population. Thus for a population of N observations (elements) and with μ, denoting the population mean, the formula for population variance . However, in almost all applications of statistics, the data being analyzed is a sample data. As a result, population variance is rarely determined. Instead, we compute a sample variance to estimate population variance, $\boldsymbol{\sigma}^2$.

10) What is Newton's Divided Difference Formula

Ans:- Consider the function y_x for the arguments $x, a, b, c, d \ldots j, k$. Then We continue this process until nth differences are reached. Assuming that y_x is represented by an nth
degree polynomial, all higher differences vanish and we have Newton's divided difference formula:
where there are $(n + 1)$ arguments $a, b, c, \ldots, k$ and $A = (x - a)$, $B = (x - b)$, $C = (x - c)$, …, $K = (x - k)$.

Corollary If the arguments $a, b, c, \ldots$ are taken as 0, 1, 2, … then and we obtain from equation

LAGRANGE'S INTERPOLATION FORMULA

In the inverse interpolation for a given value of y, the corresponding value of x is to be found. Interchanging the roles of x and $y = f(x)$ in Lagrange's interpolation formula we get Lagrange's inverse interpolation formula as

Newton's divided differences formula is

$$y = f(x) = f(x_0) + (x - x_0)\,[x_0, x_1] + (x - x_0)(x - x_1) \cdot [x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) \cdot [x_0, x_1, x_2] + \cdots = 48 + 52(x - 4) + 15(x - 4)(x - 5) + 1(x - 4)(x - 5)(x - 7) = x^2(x - 1)$$

$$\therefore f(2) = 2^2(2 - 1) = 4$$

$$f(8) = 8^2(8 - 1) = 448$$

11) What is Numerical Differentiation

Ans:-Numerical differentiation is a process of computing the derivative of a function at some assigned value of x from a given set of data

$(x_i\ f_i)\ f_i = f(x_i)\ (i = 0, 1, 2, \ldots, n)$

MAXIMUM AND MINIMUM VALUES OF A TABULATED FUNCTION

We know that maximum and minimum values of a function are found by equating the first derivative to zero and solving for the variable. We can apply this procedure to a tabulated function as well.

Differentiating Newton's Forward interpolation formula

we have neglecting higher-order terms.

For an extreme value of y, we must have $dy/dp = 0$. This yields the quadratic in p:

$$a_0 + a_1 p + a_2 p^2 = 0$$

with The values of x are found from $x = x_0 + ph$.

NUMERICAL INTEGRATION: INTRODUCTION

We can find the integral of a function $y = f(x)$ defined and continuous on an interval $[a, b]$ if there exists a function $F(x)$ such that $F'(x) = f(x)$. According to the Fundamental Theorem of Integral Calculus we

Geometrically, it represents the area under the curve $y = f(x)$ and between the ordinates $x = a$ and $x = b$.

In applications, evaluation of the integral may prove to be very complicated and not practical:

1. when we cannot find the anti derivative $F(x)$ of $f(x)$ and
2. when the integrand $f(x)$ is a tabulated function.

In such cases we have to resort to numerical evaluation which is also called mechanical quadrature. The basic idea is to replace $f(x)$ by an interpolating polynomial $\phi(x)$ using a suitable interpolation formula. We derive a general formula for numerical integration using Newton's Forward difference formula.

12) Describe Simpson's Method

Ans:-In order to find $y(x_n)$ where $x_n = x_0 + nh$ by Milne's method we proceed in the following way.

Since the value $y_0 = y(x_0)$ is given to us, we compute $y_1 = y(x_1) = y(x_0 + h)$, $y_2 = y(x_2) = y(x_0 + 2h)$, $y_3 = y(x_3) = y(x_0 + 3h)$ by Picard's or Taylor's method. Next we calculate

To evaluate: $\int_{x_0}^{x_n} f(x)dx$

1. Divide $[x_0, x_n]$ into n segments (n ³ 1)
2. Within each segment approximate $f(x)$ by an mth order polynomial,

$$p_m(x) = a_0 + a_1x + a_2x^2 + \ldots + a_mx^m$$

The polynomial order need not be the same for all segments. Then:

$$\int_{x0}^{xn} f(x)dx = \int_{x0}^{x1} pm_1(x)dx + \int_{x0}^{x2} pm_2(x)dx + \ldots + \int_{x0}^{xn} pm_n(x)dx$$

the m_i's may be the same or different. Integrate each polynomial exactly.

Order m of polynomial $p_m(x)$ determines Newton-Cotes formulas:

m	Polynomial	Formula	Error
1	linear	Trapezoid	$O(h^2)$
2	quadratic	Simpson's 1/3	$O(h^4)$
3	cubic	Simpson's 3/8	$O(h^4)$
4	quartic	Boole's Rule	$O(h^6)$
5	quintic	Boole's Rule	$O(h^6)$

13) Explain Trapezoid Rule

Ans:- For each segment (or one segment), let $f(x) \approx p_1(x) = a_0 + a_1 x$

a. Determine $a_0 + a_1 x$ from Newton DD Polynomial:

$$p_1(x) = f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}(x - x_{i-1})$$

b. Integrating [use trapezoid area formula, C&C 4th ed., Box 21.1]:

$$\int_{x_{i-1}}^{x_i} p_1(x)dx = (x_i - x_{i-1}) \frac{f(x_i) + f(x_{i-1})}{2}$$

c. Truncation Error [C&C 4th ed., Box 21.2]:

$$\int_{x_{i-1}}^{x_i} \varepsilon(x)dx = (x_i - x_{i-1}) \frac{f(x_i) + f(x_{i-1})}{2} - \frac{1}{12}(x_i - x_{i-1})^3 f''(x)$$

Integrates a linear function correctly: $f''(x) = 0$.

Easily derived by considering Taylor Series.

14) Describe Gauss Quadrature

Ans-:

Requires: $f(x)$ to be explicitly known so we can pick any x_i

Approach: $$\int_{-1}^{+1} f(x)dx = \sum_{i=0}^{n-1} w_i f(x_i) + R_n$$

where: w_i = weighting factors

x_i = sampling points selected optimally

R_n = truncation error

Pick points & weights cleverly to integrate a polynomial of order (2n - 1) exactly.

For **n = 2** Gauss Quadrature will be accurate for **cubics**.

Trapezoidal Rule is accurate for linear functions.

For **n=3** exact result for polynomials of order up to and including 5

With 3 points we want exact results for polynomials of order 5 over the interval [-1, +1].

15) **Gauss Quadrature Truncation Error**

In general, n sampling points will provide an exact solution for a 2n-1 order polynomial. With n = # of sampling points

$$R_n = \left[\frac{2^{2n+1}}{(2n+1)}\right]\left[\frac{(n!)^4}{(2n)!^3}\right] f^{(2n)}(x)$$

(similar to C&C except their n = # pts.-1)

Because (b-a) = h, the error with the composite Gauss rule is $O(h^{2n})$ globally

with n = # of pts..

This shows a superiority of order over the Newton-Cotes formulas.

16.How to improve the accuracy of Integration

Ans:-An intuitive solution is to improve the accuracy of f (x) by using a better interpolating polynomial say a quadratic

polynomial $p_2(x)$ instead. Let c = $\frac{a+b}{2}$ and h = $\frac{b\ a}{2}$ then, the

quadratic polynomial is

$$p_2(x) = \frac{(x\ \ c)(x\ \ b)}{(a\ \ c)(a\ \ b)} f(a) + \frac{(x\ \ a)(x\ \ b)}{(c\ \ a)(c\ \ b)} f(c) + \frac{(x\ \ a)(x\ \ c)}{(b\ \ a)(b\ \ c)} f(b):$$

$$\int_a^b f(x)\,dx \approx \int_a^b p_2(x)\,dx = \frac{h}{3}\left[f(a) + 4f(c) + f(b)\right] \equiv S_2(f):$$

This is called Simpson's rule.

17) Explain Interpolation ,Extrapolation & Farward difference

Ans:- Interpolation is the technique of estimating the value of a function for any intermediate value of the independent variable, while the process of computing the value of the function outside the given range is called **extrapolation**.

Forward Differences : The differences y1 – y0, y2 – y1, y3 – y2, ……, yn – yn–1 when denoted by dy0, dy1, dy2, ……, dyn–1 are respectively, called the first forward differences. Thus the first forward differences are :

Forward difference table

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
x_0	y_0					
		Δy_0				
x_1	y_1		$\Delta^2 y_0$			
$(= x_0 + h)$		Δy_1		$\Delta^3 y_0$		
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
$(= x_0 + 2h)$		Δy_2		$\Delta^3 y_1$		$\Delta^5 y_0$
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	
$= (x_0 + 3h)$		Δy_3		$\Delta^3 y_2$		
x_4	y_4		$\Delta^2 y_3$			
$= (x_0 + 4h)$		Δy_4				
x_5	y_5					
$= (x_0 + 5h)$						

18) Find the Population in 1925 of given data.

Input : Population in 1925

Year (x):	*1891*	*1901*	*1911*	*1921*	*1931*
Population (y): (in thousands)	*46*	*66*	*81*	*93*	*101*

Ans:- Output :

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1891	46				
		20			
1901	66		– 5		
		15		2	
1911	81		– 3		– 3
		12		– 1	
1921	93		– 4		
		8			
1931	101				

Value in 1925 is 96.8368

19) Explain Newton's Forward Difference Formula

Ans:-Newton's forward difference formula is a finite difference identity giving an interpolated value between tabulated points $\{f_p\}$ in terms of the first value f_0 and the powers of the forward difference Δ. For $a \in [0, 1]$, the formula states

$$f_a = f_0 + a\,\Delta + \tfrac{1}{2!}\,a\,(a-1)\,\Delta^2 + \tfrac{1}{3!}\,a\,(a-1)\,(a-2)\,\Delta^3 + \ldots. \qquad (1)$$

When written in the form

$$f\,(x+a) = \sum_{n=0}^{\infty} \frac{(a)_n\,\Delta^n\,f\,(x)}{n!}$$

with $(a)_n$ the falling factorial, the formula looks suspiciously like a finite analog of a Taylor series expansion. This correspondence was one of the motivating forces for the development of umbral calculus.

An alternate form of this equation using binomial coefficients is

$$f\,(x+a) = \sum_{n=0}^{\infty} \binom{a}{n} \Delta^n\,f\,(x),$$

where the binomial coefficient $\binom{a}{n}$ represents a polynomial of degree n in a.

The derivative of Newton's forward difference formula gives Markoff's formulas.

20) Explain Newton's Interpolation Formulae

Ans:-

As stated earlier, interpolation is the process of approximating a given function, whose values are known at $N+1$ tabular points, by a suitable polynomial, $P_N(x)$, of degree N which takes the values y_i at $x = x_i$ for $i = 0, 1, \ldots, N$. Note that if the given data has errors, it will also be reflected in the polynomial so obtained.

In the following, we shall use forward and backward differences to obtain polynomial function approximating $y = f(x)$, when the tabular points x_i's are equally spaced. Let

$$f(x) \approx P_N(x),$$

where the polynomial $P_N(x)$ is given in the following form:

$$P_N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_k(x - x_0)(x - x_1)\cdots(x - x_{k-1}) + a_N(x - x_0)(x - x_1)\cdots(x - x_{N-1}). \quad (11.4.1)$$

for some constants $a_0, a_1, \ldots a_N,$ to be determined using the fact that $P_N(x_i) = y_i$ for $i = 0, 1, \ldots, N.$

So, for $i = 0,$ substitute $x = x_0$ in (11.4.1) to get $P_N(x_0) = y_0.$ This gives us $a_0 = y_0.$ Next,

$$P_N(x_1) = y_1 \Rightarrow y_1 = a_0 + (x_1 - x_0)a_1.$$

$$a_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}.$$

So, For $i = 2, \quad y_2 = a_0 + (x_2 - x_0)a_1 + (x_2 - x_1)(x_2 - x_0)a_2,$ or equivalently

$$2h^2 a_2 = y_2 - y_0 - 2h\left(\frac{\Delta y_0}{h}\right) = y_2 - 2y_1 + y_0 = \Delta^2 y_0.$$

$$a_2 = \frac{\Delta^2 y_0}{2h^2}.$$

Thus, Now, using mathematical induction, we get

$$a_k = \frac{\Delta^k y_0}{k!\, h^k} \quad \text{for} \quad k = 0, 1, 2, \ldots, N.$$

Thus,

$$P_N(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!\, h^2}(x - x_0)(x - x_1) + \cdots + \frac{\Delta^k y_0}{k!\, h^k}(x - x_0)\cdots(x - x_{k-1}) + \frac{\Delta^N y_0}{N!\, h^N}(x - x_0)\ldots(x - x_{N-1}).$$

Web Technologies

BCA-204

Q.1 Write the basic structure of the HTML template?

Ans: The basic structure of the HTML template is:

```
<html>
    <head>
        <title></title>
    </head>
    <body>
    </body></html>
```

.

Q. 2 Name some new features which were not present in HTML but are added to HTML5?

Ans: Some new features in HTML5 include:

- DOCTYPE declaration – <!DOCTYPE html>

- section – Section tag defines a section in the document, such as a header, footer or in other sections of the document. It is used to define the structure of the document. <section></section>
- header – Header tag defines the head section of the document. A header section always sticks at the top of the document. <header></header>
- footer – Footer tag defines the footer section of the document. A footer section always sticks at the bottom of the document. <footer></footer>
- article – Article tag defines an independent piece of the content of a document. <article> </article>
- main – The main tag defines the main section in the document which contains the main content of the document. <main></main>
- figcaption – Figcaption tag defines the caption for the media element such as an image or video. <figcaption></figcaption>

Q .3 What is Anchor tag and how can you open an URL into a new tab when clicked?

Ans: Anchor tag in HTML is used for linking between two sections or two different web pages or website templates.

To open an url into a new tab in the browser upon a click, we need to add target attribute equal to _blank.

Q .4 Write an HTML code to form a table to show the below values in a tabular form with heading as Roll No., Student name, Subject Name, and values as

1, Ram, Physics

2, Shyam, Math

3, Murli, Chemistry

Ans: To represent the above values in an HTML table format, the code will be:

```
<!DOCTYPE html>

<html>

<head>

<style>
```

```
table, th, td {
border: 1px solid black;
}
</style>
</head>
<body>

<table >
<tr>
<th> Roll No. </th>
<th> Student Name </th>
<th> Subject Name </th>
</tr>
<tr>
<td> 1 </td>
<td>Ram</td>
<td> Physics </td>
</tr>
<tr>
<td> 2 </td>
```

```
<td> Shyam </td>

<td> Math </td>

</tr>

<tr>

<td> 3 </td>

<td> Murli </td>

<td> Chemistry </td>

</tr>

</table>

</body>

</html>
```

Output:

Roll No.	Student Name	Subject Name
1	Ram	Physics
2	Shyam	Math
3	Murli	Chemistry

Q.5 Define Semantic elements in HTML.

Ans: Semantic elements are HTML elements which represent its meaning to the browser and developer about its contents.

For Example – p tag represents a paragraph, a tag represents anchor tag, form tag, table tag, article tag and many more are semantic elements in HTML. Whereas div tag, span tag, bold tag are not semantic elements.

Q.6 Define attributes in HTML tag.

Ans: The HTML tag contains a field inside their tag which is called attributes of that tag.

For Example:

<img src="#"> here in this tag src is img tag attributes.

<input type=" text"> here in this tag type is input tag attributes.

Q.7 Can we modify the attribute's value of the HTML tag dynamically?

Ans: Yes, we can modify the value of the attributes by using JavaScript.

Below is the input element whose attribute will be modified from text to password, JS code to modify the attribute value:

<input type="text" id="inputField">

document.getElementById("inputField").attr("type", "password");

Q .8 How can we comment in HTML?

Ans: Comments are used by developers to keep a track of the code functionality and also help the other developers in understanding the code functionalities easily.

The commented out lines will not be shown in the browser. To comment a line, the line should start by this <!– and end by this –>. Comments can be of one line or of multiple lines.

Q.9 Enumerate the differences between Java and JavaScript?

Java is a complete programming language. In contrast, JavaScript is a coded program that can be introduced to HTML pages. These two languages are not at all inter-dependent and are designed for the different intent. Java is an object - oriented programming (OOPS) or structured programming language like C++ or C whereas JavaScript is a client-side scripting language.

Q.10 What are JavaScript Data Types?

Following are the JavaScript Data types:

- Number
- String
- Boolean
- Object
- Undefined

Q.11 . How Do I Create Frames? What Is A Frameset?

Answer :

Frames allow an author to divide a browser window into multiple (rectangular) regions. Multiple documents can be displayed in a single window, each within its own frame. Graphical browsers allow these frames to be scrolled independently of each other, and links can update the document displayed in one frame without affecting the others.

You can't just "add frames" to an existing document. Rather, you must create a frameset document that defines a particular combination of frames, and then display your content documents inside those frames. The frameset document should also include alternative non-framed content in a NOFRAMES element. The HTML 4 frames model has significant design flaws that cause usability problems for web users. Frames should be used only with great care.

Q.12 Between JavaScript and an ASP script, which is faster?

JavaScript is faster. JavaScript is a client-side language and thus it does not need the assistance of the web server to execute. On the other hand, ASP is a server-side language and hence is always slower than JavaScript. Javascript now is also a server side language (nodejs).

Q.13 How Do I Use Forms?

Answer :

The basic syntax for a form is: <FORM ACTION="[URL]">...</FORM> When the form is submitted, the form data is sent to the URL specified in the ACTION attribute. This URL should refer to a server-side (e.g., CGI) program that will process the form data. The form itself should contain

* at least one submit button (i.e., an <INPUT TYPE="submit" ...> element),
* form data elements (e.g., <INPUT>, <TEXTAREA>, and <SELECT>) as needed, and
* additional markup (e.g., identifying data elements, presenting instructions) as needed.

Q.14 How Can I Check For Errors?

Answer :

HTML validators check HTML documents against a formal definition of HTML syntax and then output a list of errors. Validation is important to give the best chance of correctness on unknown browsers (both existing browsers that you haven't seen and future browsers that haven't been written yet).

HTML checkers (linters) are also useful. These programs check documents for specific problems, including some caused by invalid markup and others caused by common browser bugs. Checkers may pass some invalid documents, and they may fail some valid ones.

All validators are functionally equivalent; while their reporting styles may vary, they will find the same errors given identical input. Different checkers are programmed to look for different problems, so their reports will vary significantly from each other. Also, some programs that are called validators (e.g. the "CSE HTML Validator") are really linters/checkers. They are still useful, but they should not be confused with real HTML validators.

When checking a site for errors for the first time, it is often useful to identify common problems that occur repeatedly in your markup. Fix these problems everywhere they occur (with an automated process if possible), and then go back to identify and fix the remaining problems.

Link checkers follow all the links on a site and report which ones are no longer functioning. CSS checkers report problems with CSS style sheets.

Example:

```
document.write("This is \a program");
```

And if you change to a new line when not within a string statement, then javaScript ignores break in line.

Example:

```
var x=1, y=2,
z=
x+y;
```

The above code is perfectly fine, though not advisable as it hampers debugging.

Q.15 What are undeclared and undefined variables?

Undeclared variables are those that do not exist in a program and are not declared. If the program tries to read the value of an undeclared variable, then a runtime error is encountered.

Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.

Q.16 . What is a prompt box?

A prompt box is a box which allows the user to enter input by providing a text box. Label and box will be provided to enter the text or number.

Q.17 What are escape characters?

Escape characters (Backslash) is used when working with special characters like single quotes, double quotes, apostrophes and ampersands. Place backslash before the characters to make it display.

Example:

```
document.write "I m a "good" boy"
document.write "I m a \"good\" boy"
```

Q.18 Explain how to read and write a file using JavaScript?

There are two ways to read and write a file using Javacript

- Using JavaScript extensions
- Using a web page and Active X objects

Q.19 . What are JavaScript Cookies?

Cookies are the small test files stored in a computer and it gets created when the user visits the websites to store information that they need. Example could be User Name details and shopping cart information from the previous visits.

Q.20 Explain what is pop()method in JavaScript?

The pop() method is similar as the shift() method but the difference is that the Shift method works at the start of the array. Also the pop() method take the last element off of the given array and returns it. The array on which is called is then altered.

JAVA PROGRAMMING

BCA-206

BCA-4TH SEMESTER

Q1. Give the difference between C, C++ and Java.

Basis	C	C++	Java
Year of development	C was developed in 1972	C++ was developed in 1979	Java was developed in 1991
Name of developer	C was developed by Dennis Ritchie	C++ was developed by Bjarne Stroustrup	Java was developed by James Gosling
Successor of	C was developed after BCPL	C++ was developed after C	Java was developed after c++
Programming paradigm	C follows procedural programming paradigm	C++ follows object oriented programming paradigm	Java follows object oriented paradigm
Whether platform dependent or not?	C is platform dependent	C++ is platform dependent	Java is platform independent
Purpose	C was designed basically for programming applications and for system programming.	C++ was also designed for programming applications and for system programming.	Java was designed for class based, concurrent object oriented programming.
Programming approach	C uses top down approach of programming	C++ uses bottom up approach of programming	Java uses bottom up approach of programming
Number of keywords allowed	C language has 32 keywords	C++ language has 63 keywords	Java language has 50 keywords
Union and Structure	C supports Union and	C++ supports Union	Java neither supports

	Structure both	and Structure both	C nor C++
Concept of inheritance	C does not have any such concept	C++ supports various types of Inheritance	Java supports inheritance except multiple inheritance
Use of Pointers	C supports use of pointer variables	C++ also supports use of pointer variables	Java does not support use of pointer variables
Method of translating source code	C is a compiled language. Source code is compiled	C++ is also a compiled language. Source code is compiled.	Java is both compiled and interpreted language.
Method of storage allocation	C uses malloc and calloc for allocating storage	C++ uses new and delete to allocate storage.	Java uses garbage collector
Concept of constructors	No such concept	C++ supports constructors	Java supports constructors
Database connection	C does not provide mechanism for database connection	C++ does not provide any mechanism for database connection	Java provides mechanism for database connection
Class	C does not make use of classes	C++ makes use of classes	Java makes use of classes
Use of header files	C makes use of predefined header files	C++ makes use of header files	Java makes use of packages and not header files.
Applets	C does not support use of internet programming method such as Applet	C++ does not support use of internet programming method such as Applet	Java supports use of applets for the purpose of internet programming.

Use of Exception handling	C does uses exception handling for exception generated while execution of program.	C++ makes use of exception handling for handling various types of exceptions generated while execution of program.	Java makes use of exception handling in a very effective way to handle exceptions generated while execution of program.
Operator overloading	There is no such concept of operator overloading in C	C++ uses concept of operator overloading	There is no such concept of operator overloading in Java
Use of templates	There are no templates in C	C++ allows generic programming through use of templates	There are no templates in Java
Goto keyword	It supports Goto keyword	It supports Goto keyword	It does not uses Goto keyword
Including files and statements	C makes use of #include pre processor directive to include other files	C++ uses #include pre processor directive to include other files	Java makes use of import statement to include other files.
Enumerated data type	C supports use of enum data type	C++ supports use of enum data type	Java does not support use of enum data type

Q2. State any six features of Java in detail.

Ans. **1. Java is Simple:**

The Java programming language is easy to learn. Java code is easy to read and write.

2. Java is Familiar:

Java is similar to C/C++ but it removes the drawbacks and complexities of C/C++ like pointers and multiple inheritances. So if you have background in C/C++, you will find Java familiar and easy to learn.

3. Java is an Object-Oriented programming language:

Unlike C++ which is semi object-oriented, Java is a fully object-oriented programming language. It has all OOP features such as abstraction, encapsulation, inheritance and polymorphism.

4. Java is Robust:

With automatic garbage collection and simple memory management model (no pointers like C/C++), plus language features like generics, try-with-resources. Java guides programmer toward reliable programming habits for creating highly reliable applications.

5. Java is Secure:

The Java platform is designed with security features built into the language and runtime system such as static type-checking at compile time and runtime checking (security manager), which let you creating applications that can't be invaded from outside. You never hear about viruses attacking Java applications.

6. Java is High Performance:

Java code is compiled into byte code which is highly optimized by the Java compiler, so that the Java virtual machine (JVM) can execute Java applications at full speed. In addition, compute-intensive code can be re-written in native code and interfaced with Java platform via *Java Native Interface* (JNI) thus improve the performance.

Q3. What is the need of an interface in Java?

Ans. There are several reasons, an application developer needs an interface, one of them is Java's feature to provide multiple inheritances at interface level. It allows you to write flexible code, which can adapt to handle future requirements. Some of the concrete reasons why we should use interfaces are-

1) If you only implement methods in subclasses, the callers will not be able to call them via

the interface (not common point where they are defined).

2) Java 8 will introduce default implementation of methods inside the interface, but that should be used as exception rather than rule. Even Java designer used in that way, it was introduced to maintain backward compatibility along with supporting lambda expression. All evolution of Stream API was possible due to this change.

3) Interfaces are a way to declare a contract for implementing classes to fulfil; it's the primary tool to create abstraction and decoupled designs between consumers and producers.

4) Because of multiple inheritance, interface allows you to treat one thing differently. For example a class can be treated as Canvas during drawing and EventListener during event processing. Without interface, it's not possible for a class to behave like two different entity at two different situations.

5) *"Programming to interface than implementation"* is one of the popular Object oriented design principle, and use of interface promotes this. A code written on interface is much more flexible than the one which is written on implementation.

6) Use of interface allows you to supply a new implementation, which could be more robust, more performance in later stage of your development.

Q4. What is Inheritance? Explain different types of inheritance in Java with suitable diagram and small segment of codes.

Ans. Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.

The syntax of Java Inheritance

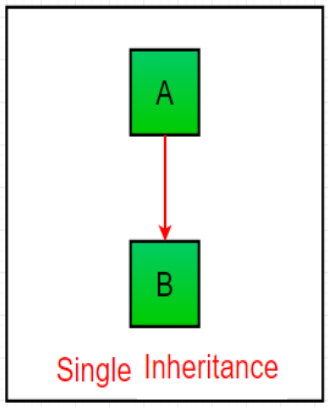
```
class Subclass-name extends Superclass-name
{
  //methods and fields
}
```

The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

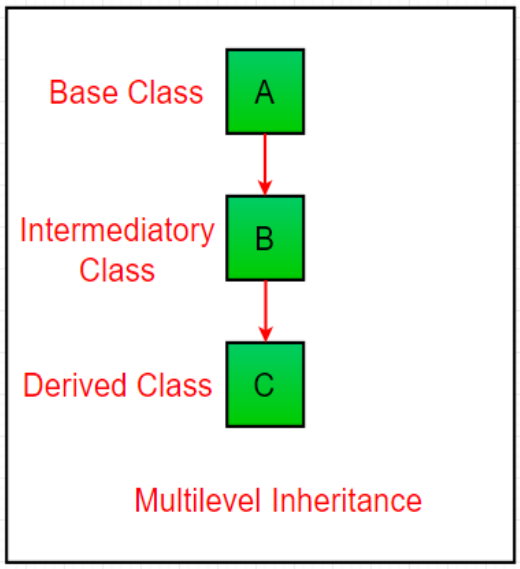
In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.

Types of Inheritance in Java

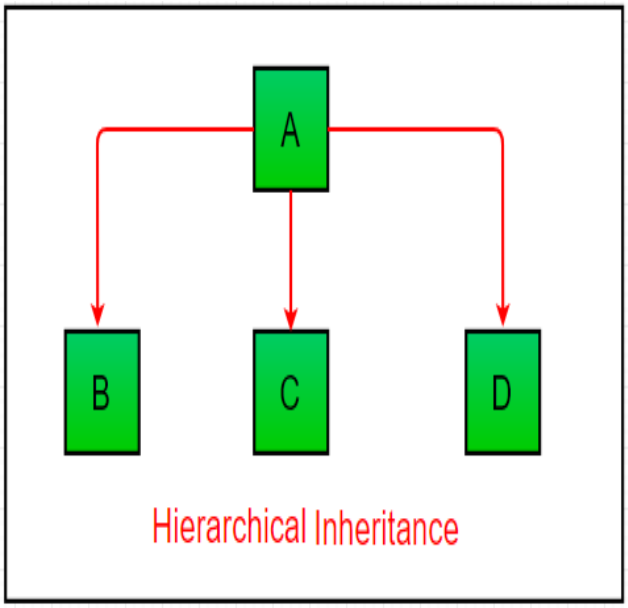
1. **Single Inheritance:** In single inheritance, subclasses inherit the features of one superclass. In image below, the class A serves as a base class for the derived class B.



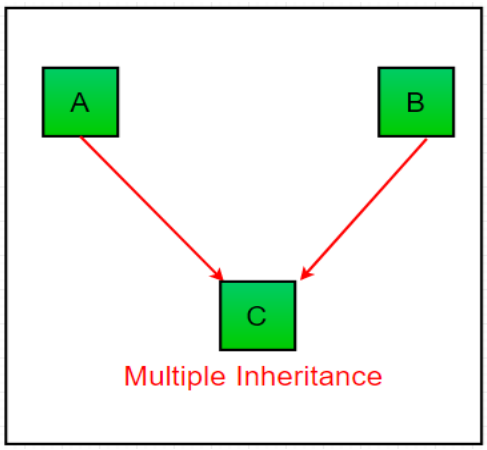
2. **Multilevel Inheritance:** In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also act as the base class to other class. In below image, the class A serves as a base class for the derived class B, which in turn serves as a base class for the derived class C. In Java, a class cannot directly access the grandparent's members.



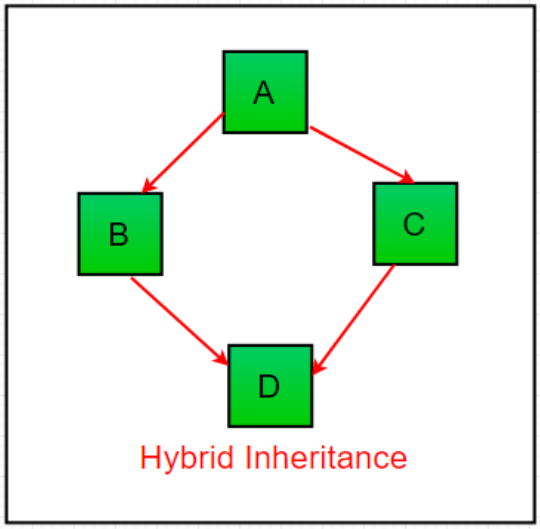
3. **Hierarchical Inheritance:** In Hierarchical Inheritance, one class serves as a superclass (base class) for more than one sub class. In below image, the class A serves as a base class for the derived class B,C and D.



4. **Multiple Inheritance (Through Interfaces):** In Multiple inheritance, one class can have more than one superclass and inherit features from all parent classes. Please note that Java does **not** support multiple inheritances with classes. In java, we can achieve multiple inheritances only through Interfaces. In image below, Class C is derived from interface A and B.



5. **Hybrid Inheritance (Through Interfaces) :** It is a mix of two or more of the above types of inheritance. Since java doesn't support multiple inheritance with classes, the hybrid inheritance is also not possible with classes. In java, we can achieve hybrid inheritance only through Interfaces.



Q5. How is method overloading different from method overriding?

No.	Method Overloading	Method Overriding

1)	Method overloading is used *to increase the readability* of the program.	Method overriding is used *to provide the specific implementation* of the method that is already provided by its super class.
2)	Method overloading is performed *within class*.	Method overriding occurs *in two classes* that have IS-A (inheritance) relationship.
3)	In case of method overloading, *parameter must be different*.	In case of method overriding, *parameter must be same*.
4)	Method overloading is the example of *compile time polymorphism*.	Method overriding is the example of *run time polymorphism*.
5)	In java, method overloading can't be performed by changing return type of the method only. *Return type can be same or different* in method overloading. But you must have to change the parameter.	*Return type must be same or covariant* in method overriding.

Q6. Explain different access modifiers available in Java?

Ans. The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

There are four types of Java access modifiers:

1. **Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
2. **Default**: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

3. **Protected**: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.
4. **Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

Q7. Difference between checked and unchecked exceptions with the help of an example.

Ans. 1) Checked: are the exceptions that are checked at compile time. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using *throws* keyword.

For example, consider the following Java program that opens file at location "C:\test\a.txt" and prints the first three lines of it. The program doesn't compile, because the function main() uses FileReader() and FileReader() throws a checked exception *FileNotFoundException*. It also uses readLine() and close() methods, and these methods also throw checked exception *IOException*

```
import java.io.*;

class Main {
   public static void main(String[] args) {
      FileReader file = new FileReader("C:\\test\\a.txt");
      BufferedReader fileInput = new BufferedReader(file);

      // Print first 3 lines of file "C:\test\a.txt"
      for (int counter = 0; counter < 3; counter++)
         System.out.println(fileInput.readLine());

      fileInput.close();
   }
}
```

Output:

```
Exception in thread "main" java.lang.RuntimeException: Uncompilable source code -
```

```
unreported exception java.io.FileNotFoundException; must be caught or declared to be
thrown
```

2) Unchecked are the exceptions that are not checked at compiled time. In C++, all exceptions are unchecked, so it is not forced by the compiler to either handle or specify the exception. It is up to the programmers to be civilized, and specify or catch the exceptions. In Java exceptions under *Error* and *RuntimeException* classes are unchecked exceptions, everything else under throwable is checked.

Consider the following Java program. It compiles fine, but it throws *ArithmeticException* when run. The compiler allows it to compile, because *ArithmeticException* is an unchecked exception.

```
class Main {
  public static void main(String args[]) {
    int x = 0;
    int y = 10;
    int z = y/x;
  }
}
```

Output:

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
   at Main.main(Main.java:5)
Java Result: 1
```

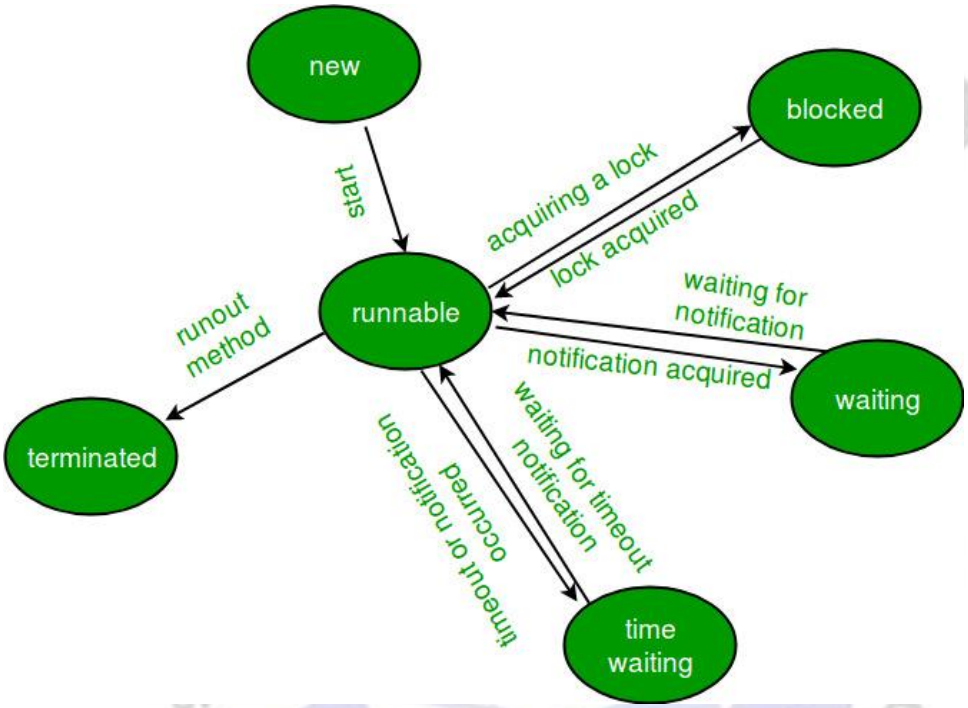
Q8. Explain the life cycle of a thread. Describe any five methods from thread cycle.

Ans. A thread in Java at any point of time exists in any one of the following states. A thread lies only in one of the shown states at any instant:

1. New
2. Runnable
3. Blocked

4. Waiting
5. Timed Waiting
6. Terminated

The diagram shown below represent various states of a thread at any instant of time.



Life Cycle of a thread

1. **New Thread:** When a new thread is created, it is in the new state. The thread has not yet started to run when thread is in this state. When a thread lies in the new state, it's code is yet to be run and hasn't started to execute.
2. **Runnable State:** A thread that is ready to run is moved to runnable state. In this state, a thread might actually be running or it might be ready run at any instant of time. It is the responsibility of the thread scheduler to give the thread, time to run. A multi-threaded program allocates a fixed amount of time to each individual thread. Each and every thread runs for a short while and then pauses and relinquishes the CPU to another thread, so that other threads can get a chance to run. When this happens, all such threads that are ready to run, waiting for the CPU and the currently running thread lies in runnable state.

3. **Blocked/Waiting state:** When a thread is temporarily inactive, then it's in one of the following states:
    - Blocked
    - Waiting

    For example, when a thread is waiting for I/O to complete, it lies in the blocked state. It's the responsibility of the thread scheduler to reactivate and schedule a blocked/waiting thread. A thread in this state cannot continue its execution any further until it is moved to runnable state. Any thread in these states does not consume any CPU cycle.

    A thread is in the blocked state when it tries to access a protected section of code that is currently locked by some other thread. When the protected section is unlocked, the schedule picks one of the thread which is blocked for that section and moves it to the runnable state. Whereas, a thread is in the waiting state when it waits for another thread on a condition. When this condition is fulfilled, the scheduler is notified and the waiting thread is moved to runnable state.

    If a currently running thread is moved to blocked/waiting state, another thread in the runnable state is scheduled by the thread scheduler to run. It is the responsibility of thread scheduler to determine which thread to run.

4. **Timed Waiting:** A thread lies in timed waiting state when it calls a method with a time out parameter. A thread lies in this state until the timeout is completed or until a notification is received. For example, when a thread calls sleep or a conditional wait, it is moved to a timed waiting state.
5. **Terminated State:** A thread terminates because of either of the following reasons:
    - Because it exists normally. This happens when the code of thread has entirely executed by the program.
    - Because there occurred some unusual erroneous event, like segmentation fault or an unhandled exception.

    A thread that lies in a terminated state does no longer consumes any cycles of CPU.

Q9. Explain any five string class methods in detail.

Ans.

Method	Description	Return Type
charAt()	Returns the character at the specified index (position)	char
codePointAt()	Returns the Unicode of the character at the specified index	int
codePointBefore()	Returns the Unicode of the character before the specified index	int
codePointCount()	Returns the Unicode in the specified text range of this String	int
compareTo()	Compares two strings lexicographically	int

Q10. What is an exception? Explain exception handling in java with example.

Ans. An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

Q11. What is the role of priorities in multithreading?

Ans. In a Multi threading environment, thread scheduler assigns processor to a thread based on priority of thread. Whenever we create a thread in Java, it always has some priority assigned to it. Priority can either be given by JVM while creating the thread or it can be given by programmer explicitly.
Accepted value of priority for a thread is in range of 1 to 10. There are 3 static variables defined in Thread class for priority.

public static int MIN_PRIORITY: This is minimum priority that a thread can have. Value for this is 1.

public static int NORM_PRIORITY: This is default priority of a thread if do not explicitly define it. Value for this is 5.

public static int MAX_PRIORITY: This is maximum priority of a thread. Value for this is 10.

Q12. Explain applet life cycle.

Ans. Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Lifecycle of Java Applet

1. Applet is initialized.
2. Applet is started.
3. Applet is painted.
4. Applet is stopped.
5. Applet is destroyed.

Below is the description of each applet life cycle method:

1. **init():** The init() method is the first method to execute when the applet is executed. Variable declaration and initialization operations are performed in this method.

2. **start():** The start() method contains the actual code of the applet that should run. The start() method executes immediately after the *init()* method. It also executes whenever

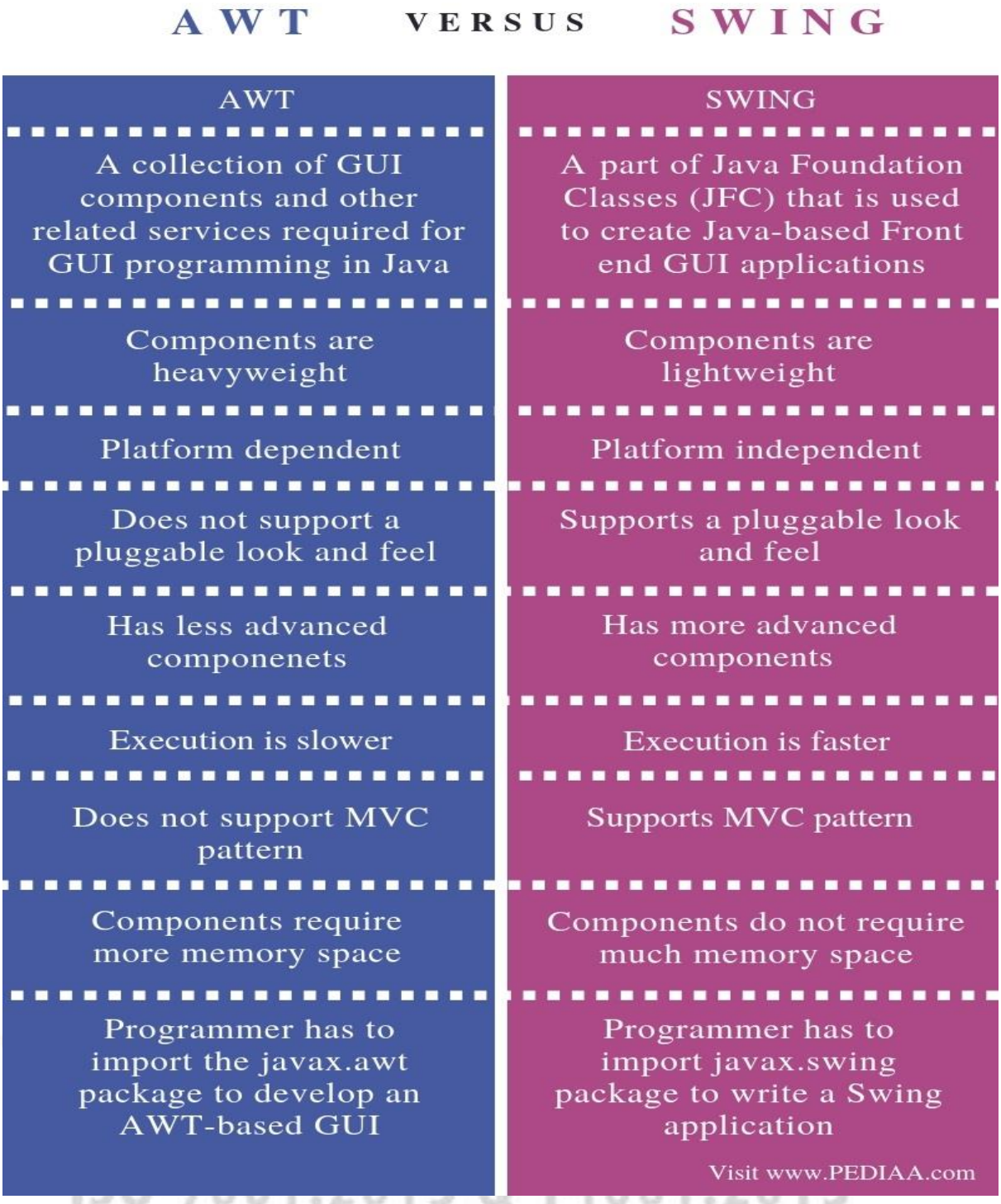
the applet is restored, maximized or moving from one tab to another tab in the browser.

3. **paint():** The paint() method is used to redraw the output on the applet display area. The paint() method executes after the execution of *start()* method and whenever the applet or browser is resized.

4. **stop():** The stop() method stops the execution of the applet. The stop() method executes when the applet is minimized or when moving from one tab to another in the browser.

5. **destroy():** The destroy() method executes when the applet window is closed or when the tab containing the webpage is closed. *stop()* method executes just before when destroy() method is invoked. The destroy() method removes the applet object from memory.

Q13. Difference between swings and AWT.



Q14. What is delegation event model?

Ans. The event model is based on the Event Source and Event Listeners. Event Listener is an object that receives the messages / events. The Event Source is any object which creates the message / event. The Event Delegation model is based on – The Event Classes, The Event

Listeners, Event Objects.

There are three participants in event delegation model in Java;
- Event Source – the class which broadcasts the events
- Event Listeners – the classes which receive notifications of events
- Event Object – the class object which describes the event.

An event occurs (like mouse click, key press, etc) which is followed by the event is broadcasted by the event source by invoking an agreed method on all event listeners. The event object is passed as argument to the agreed-upon method. Later the event listeners respond as they fit, like submit a form, displaying a message / alert etc.

Q15. Discuss various layout managers available in AWT.

Ans. The layout manager automatically positions all the components within the container. If we do not use layout manager then also the components are positioned by the default layout manager. It is possible to layout the controls by hand but it becomes very difficult because of the following two reasons.

- It is very tedious to handle a large number of controls within the container.
- Oftenly the width and height information of a component is not given when we need to arrange them.

Java provides us with various layout manager to position the controls. The properties like size, shape and arrangement varies from one layout manager to other layout manager. When the size of the applet or the application window changes the size, shape and arrangement of the components also changes in response i.e. the layout managers adapt to the dimensions of appletviewer or the application window.

There are following classes that represents the layout managers:

1. java.awt.BorderLayout
2. java.awt.FlowLayout
3. java.awt.GridLayout
4. java.awt.CardLayout

5. java.awt.GridBagLayout
6. javax.swing.BoxLayout
7. javax.swing.GroupLayout
8. javax.swing.ScrollPaneLayout
9. javax.swing.SpringLayout etc.

Q16. Explain: 1) Adapter class 2) Inner class

Ans. **Adapter Classes**

Java adapter classes provide the default implementation of listener *interfaces.* If you inherit the adapter class, you will not be forced to provide the implementation of all the methods of listener interfaces. So it saves code.

The adapter classes are found
in **java.awt.event**, **java.awt.dnd** and **javax.swing.event** packages.

Inner Classes

Java inner class or nested class is a class which is declared inside the class or interface. We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable. Additionally, it can access all the members of outer class including private data members and methods.

Syntax of Inner class

```
class Java_Outer_class{
 //code
 class Java_Inner_class{
  //code
 }
}
```

Advantage of java inner classes

There are basically three advantages of inner classes in java. They are as follows:

1) Nested classes represent a special type of relationship that is **it can access all the members (data members and methods) of outer class** including private.

2) Nested classes are used **to develop more readable and maintainable code** because it logically group classes and interfaces in one place only.

3) **Code Optimization**: It requires less code to write.

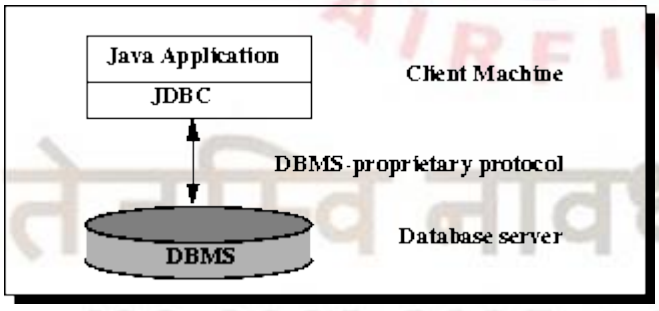
Q17. What is JDBC? Discuss the general JDBC architecture?

Ans. JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition).

JDBC Architecture

The JDBC API supports both two-tier and three-tier processing models for database access.

Figure 1: Two-tier Architecture for Data Access.

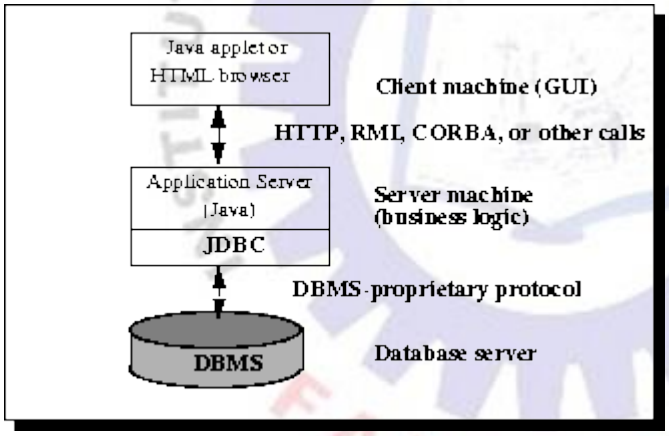


In the two-tier model, a Java application talks directly to the data source. This requires a JDBC driver that can communicate with the particular data source being accessed. A user's commands are delivered to the database or other data source, and the results of those statements are sent back to the user. The data source may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the data source as the server.

The network can be an intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then sends the commands to the data source. The data source processes the commands and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that it simplifies the deployment of applications. Finally, in many cases, the three-tier architecture can provide performance advantages.

Figure 2: Three-tier Architecture for Data Access.



Q18. Explain JDBC drivers?

Ans. JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver

1) JDBC-ODBC bridge driver

The JDBC-ODBC bridge driver uses ODBC driver to connect to the database. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC function calls. This is now discouraged because of thin driver.

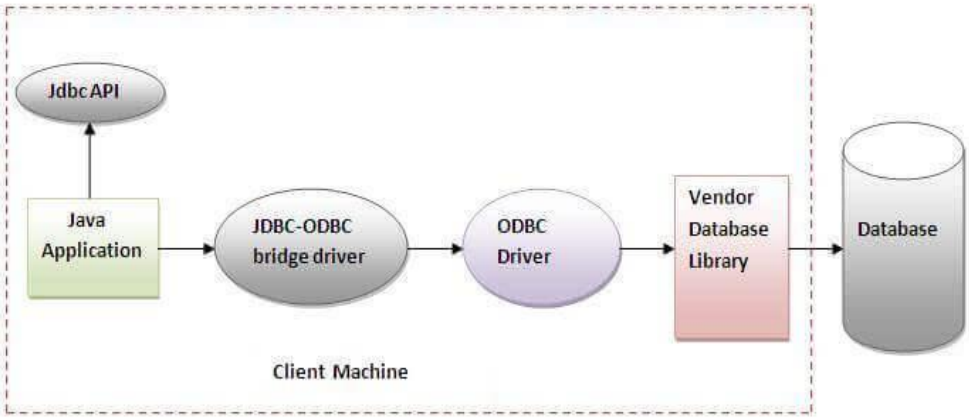


Figure- JDBC-ODBC Bridge Driver

Oracle does not support the JDBC-ODBC Bridge from Java 8. Oracle recommends that you use JDBC drivers provided by the vendor of your database instead of the JDBC-ODBC Bridge.

Advantages:

- easy to use.
- can be easily connected to any database.

Disadvantages:

- Performance degraded because JDBC method call is converted into the ODBC function calls.
- The ODBC driver needs to be installed on the client machine.

2) Native-API driver

The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in java.

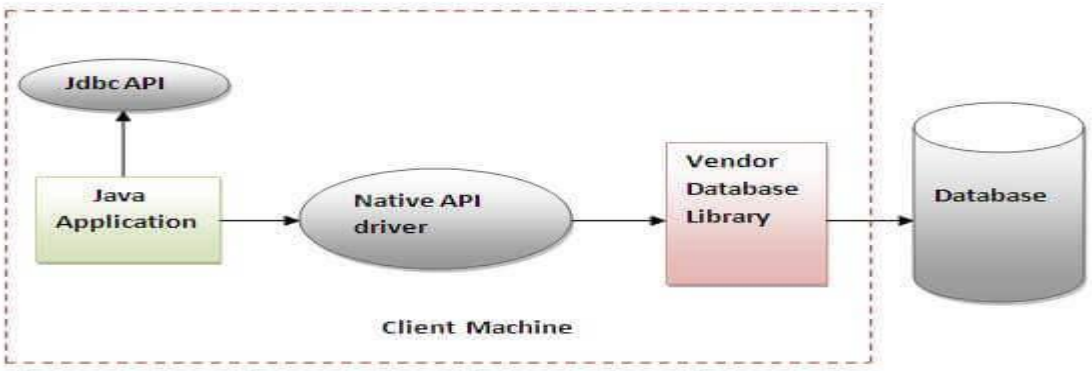


Figure- Native API Driver

Advantage:

- performance upgraded than JDBC-ODBC bridge driver.

Disadvantage:

- The Native driver needs to be installed on the each client machine.
- The Vendor client library needs to be installed on client machine.

3) Network Protocol driver

The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol. It is fully written in java.

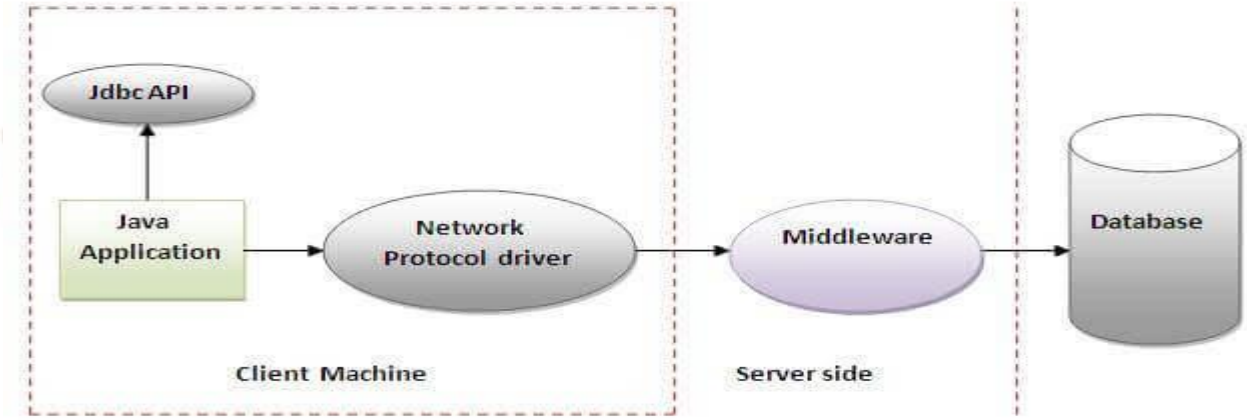


Figure- Network Protocol Driver

Advantage:

- No client side library is required because of application server that can perform many tasks like auditing, load balancing, logging etc.

Disadvantages:

- Network support is required on client machine.
- Requires database-specific coding to be done in the middle tier.
- Maintenance of Network Protocol driver becomes costly because it requires database-specific coding to be done in the middle tier.

4) Thin driver

The thin driver converts JDBC calls directly into the vendor-specific database protocol. That is why it is known as thin driver. It is fully written in Java language.

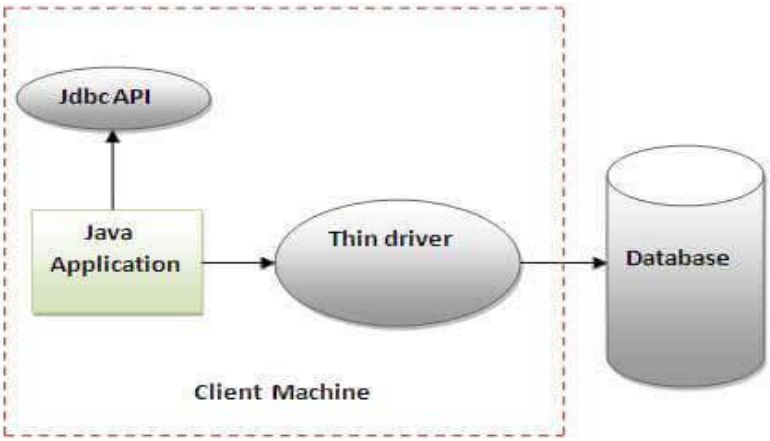


Figure- Thin Driver

Advantage:

- Better performance than all other drivers.
- No software is required at client side or server side.

Disadvantage:

- Drivers depend on the Database.

Q19. Explain servlet life cycle and its interfaces.

Ans. A servlet life cycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servlet.

- The servlet is initialized by calling the **init()** method.
- The servlet calls **service()** method to process a client's request.
- The servlet is terminated by calling the **destroy()** method.
- Finally, servlet is garbage collected by the garbage collector of the JVM.

The init() Method

The init method is called only once. It is called only when the servlet is created, and not called for any user requests afterwards. So, it is used for one-time initializations, just as with the init method of applets.

The servlet is normally created when a user first invokes a URL corresponding to the servlet, but you can also specify that the servlet be loaded when the server is first started.

When a user invokes a servlet, a single instance of each servlet gets created, with each user request resulting in a new thread that is handed off to doGet or doPost as appropriate. The init() method simply creates or loads some data that will be used throughout the life of the servlet.

The init method definition looks like this −

```
public void init() throws ServletException {
  // Initialization code...
}
```

The service() Method

The service() method is the main method to perform the actual task. The servlet container (i.e. web server) calls the service() method to handle requests coming from the client(browsers) and to write the formatted response back to the client.

Each time the server receives a request for a servlet, the server spawns a new thread and calls service. The service() method checks the HTTP request type (GET, POST, PUT, DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

Here is the signature of this method −

```
public void service(ServletRequest request, ServletResponse response)
  throws ServletException, IOException {
}
```

The destroy() Method

The destroy() method is called only once at the end of the life cycle of a servlet. This method gives your servlet a chance to close database connections, halt background threads, write cookie lists or hit counts to disk, and perform other such cleanup activities.

After the destroy() method is called, the servlet object is marked for garbage collection. The destroy method definition looks like this −

```
public void destroy() {
  // Finalization code...
}
```

Some Important Classes and Interfaces of javax.servlet

INTERFACES	CLASSES
Servlet	ServletInputStream
ServletContext	ServletOutputStream
ServletConfig	ServletRequestWrapper

ServletRequest	ServletResponseWrapper
ServletResponse	ServletRequestEvent
ServletContextListener	ServletContextEvent
RequestDispatcher	ServletRequestAttributeEvent
SingleThreadModel	ServletContextAttributeEvent
Filter	ServletException
FilterConfig	UnavailableException
FilterChain	GenericServlet

Q20. Write short notes on InetAddress class and its factory methods.

Ans. Java InetAddress class represents an IP address. The java.net. InetAddress class provides methods to get the IP of any host name *for example* www.javatpoint.com, www.google.com, www.facebook.com, etc.

An IP address is represented by 32-bit or 128-bit unsigned number. An instance of InetAddress represents the IP address with its corresponding host name. There are two types of address types: Unicast and Multicast. The Unicast is an identifier for a single interface whereas Multicast is an identifier for a set of interfaces.

Moreover, InetAddress has a cache mechanism to store successful and unsuccessful host name resolutions.

Factory method is a creational design pattern which solves the problem of creating product objects without specifying their concrete classes.

Factory Method defines a method, which should be used for creating objects instead of direct constructor call (new operator). Subclasses can override this method to change the class of objects that will be created.

Q21. Write a servlet to display current date and time.

Ans. import java.io.*;
import javax.servlet.*;

```
public class DateSrv extends GenericServlet
{
    //implement service()
    public void service(ServletRequest req, ServletResponse res) throws IOException,
ServletException
    {
        //set response content type
        res.setContentType("text/html");
        //get stream obj
        PrintWriter pw = res.getWriter();
        //write req processing logic
        java.util.Date date = new java.util.Date();
        pw.println("<h2>"+"Current Date & Time: " +date.toString()+"</h2>");
        //close stream object
        pw.close();
    }
}
```

SOFTWARE ENGINEERING

BCA- 208

BCA- 4TH SEMESTER

Q1. **What is Software Development Life Cycle? (SDLC)**

Ans. System Development Life Cycle (SDLC) is the overall process of developing information systems through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance.

Q2. **Explain the different phases involved in waterfall life cycle.**

Ans. Phase I – Modeling Phase

In this phase we view the software product as part of a larger system or organization where the product is required. This is basically a system view where all the system elements are created.

Phase II – Software Requirements Analysis

Here we have a phase where the requirements are gathered. The information domain for the software is understood. The function, behaviour, performance and interfacing of the software are determined. The requirements of the software and the customer are decided upon.

Phase III – Design

This determines the data structures, the software architecture, the interface representations and the procedural (algorithmic) detail that goes into the software.

Phase IV – Code Generation

Here the actual programming is done to obtain the machine code; it is an implementation of the design.

Phase V – Testing

The testing is a process that goes hand in hand with the production of the machine code. There are a number of testing strategies. First unit testing is done and then integration testing. Alpha testing is to see if the software is as per the analysis model whereas beta

testing is to see if the software is what the customer wanted.

Phase VI – Installation

The software is released to the customer.

Phase VII - Maintenance

This is the largest phase of the software life cycle. Maintenance can be of different types: to modify the software as the requirements of the customer evolve, to remove the residual bugs in the software etc.

Q3. **What is data modeling? Give 5 examples for data modeling.**

Ans. Data modeling is the act of exploring data-oriented structures. Like other modeling artifacts data models can be used for a variety of purposes, from high-level conceptual models to physical data models. From the point of view of an object-oriented developer, data modeling is conceptually similar to class modeling. With data modeling you identify entity types whereas with class modelling you identify classes. Data attributes are assigned to entity types just as you would assign attributes and operations to classes.

Examples for data modeling include:

- Entity-Relationship diagrams
- Entity-Definition reports
- Entity and attributes report
- Table definition report
- Relationships, inheritance, composition and aggregation.

Q4. **Explain all the phases involved in the implementation phase.**

Ans. Conduct system Test

In this test software packages and in – house programs have been installed and tested, we need to conduct a final system test. All software packages, custom- built programs, and many existing programs that comprise the new system must be tested to ensure that they all work together. This task involves analysts, owners, users, and builders.

Prepare Conversion Plan

On successful completion of system test, we can begin preparations to place the new system into operation. Using the design specifications for the new system, the system analyst will develop a detailed conversion plan. This plan will identify Database to be installed, end –

user training and documentation that needed to be developed, and a strategy for converting from the old system to the new system. The conversion plan may include one of the following commonly used installation strategies

1) Abrupt Cut-over

2) Parallel Conversion

3) Location Conversion

4) Staged Conversion

Install Databases

In the previous phase we built and tested the database. To place the system into operation we need fully loaded databases. The purpose of this task is to populate the new systems databases with existing database from the old system. System builders play a primary role in this activity.

Train Users

Converting to a new system necessitates that system users be trained and provided with documentation that guides them through using the new system. Training can be performed one on one; however group training is preferred. This task will be completed by the system analysts and involves system owners and users.

Convert to New System

Conversion to the new system from old system is a significant milestone. After conversion, the ownership of the system officially transfers from the analysts and programmers to the end users. The analyst completes this task by carrying out the conversion plan Recall that the conversion plan includes detailed installation strategies to follow for converting from the existing to the new production information system. This task involves the system owners, users, analysts, designers, and builders.

Q5. **List and explain different types of testing done during the testing phase.**

Ans. Unit

Involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. Unit testing involves the use of debugging technology and testing techniques at an application component level and is typically the responsibility of the developers, not the QA staff.

Integration

As the system is integrated, it is tested by the system developer for specification compliance.

•Concerned with testing the system as it is integrated from its components

•Integration testing is normally the most expensive activity in the systems integration process

•Should focus on:

•Interface testing where the interactions between sub-systems and components are tested

•Property testing where system properties such as reliability, performance and usability are tested

System

Testing the system as a whole to validate that it meets its specification and the objectives of its users. The testing of a complete system prior to delivery. The purpose of system testing is to identify defects that will only surface when a complete system is assembled. That is, defects that cannot be attributed to individual components or the interaction between two components. System testing includes testing of performance, security, configuration sensitivity, startup and recovery from failure modes. Involves test cases designed to validate that an application and its supporting hardware/software components are properly processing business data and transactions. System testing requires the use of regression testing techniques to validate that business functions are meeting defined requirements.

Black Box

This is testing without knowledge of the internal workings of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs. It is because of this that black box testing can be considered testing with respect to the specifications, no other knowledge of the program is necessary. For this reason, the tester and the programmer can be independent of one another, avoiding programmer bias toward his own work.

White Box

Also known as *glass box, structural, clear box* and *open box testing*. White Box is a software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike Black Box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable.

Q6. **List and explain all the phases involved in the construction phase.**

Ans. Build and Test Networks

• In many cases new or enhanced applications are built around existing networks. If so there is no problem.

• However if the new application calls for new or modified networks they must normally be implemented before building and testing databases and writing or installing computer programs that will use those networks.

• This phase involves analysts, designers and builders

• A network designer and network administrator assume the primary responsibility for completing this task.

Build and Test Databases

• This task must immediately precede other programming activities because databases are the resource shared by the computer programs to be written. If new or modified databases are required for the new system, we can now build and test those databases.

• This task involves system users, analysts, designers, and builders.

• The same system specialist that designed the database will assume the primary responsibility in completing this task

Install and Test New Software Packages

• Some systems solutions may have required the purchase or lease of software packages. If so, once networks and databases for the new system have been built, we can install and test the new software.

• This activity typically involves systems analysts, Designers, builders, vendors and consultants.

Write and Test New Programs

• In this phase we are ready to develop any programs for the new system. Prototype programs are frequently constructed in the design phase. However, these prototypes are rarely fully functional or incomplete.

• This task involves the system analysts, designers and builders.

Q7. **What is data conversion? Why is it necessary?**

Ans. Data Conversion is the changing of the data structure to accommodate new or different needs for the data. Different operating systems have different application software, and each application normally has its own internal way of saving data. There are some standards such

as CSV files for databases and RTF files for word processing text, however, these are few and far between and often only save the basic information rather than the full structure.

Q8. **What is change management?**

Ans. Computer based systems are dynamic. As the business Environment changes, there is a need of some changes to the information system. The changes occur not only during the study, design, and development phases of the life cycle of the system. In this process there are two elements that are essential to the management of change.

• The performance review board, which can make management–level decisions about system modifications.

• Baseline documentation, which can be referred to, to determine the extent and impact of proposed modifications.

Q9. **What is user acceptance testing? Explain different testings in user acceptance testing. Why is it necessary?**

Ans. User Acceptance Testing is a phase of software development in which the software is tested in the "real world" by the intended audience.

Different testings are:

Alpha Testing

Alpha testing is the software prototype stage when the software is first able to run. It will not have all the intended functionality, but it will have core functions and will be able to accept inputs and generate outputs. An alpha test usually takes place in the developer's offices on a separate system.

Beta Testing

The beta phase of software design exposes a new product, which has just emerged from in-house (alpha) testing, to a large number of real people, real hardware, and real usage. Beta testing is not a method of getting free software long-term, because the software expires shortly after the testing period.

User acceptance testing is used to know if the system is working or not (both clients & in-house).

Q10. **What are functional and non-functional requirements?**

Ans. Functional

• How the system should react to the particular inputs

• How the system should behave to the particular situations

• What the system should not do

Non functional

• Constraints on the services or functions

• Time constraints

• Constraints on the development process

Q11. **Explain the steps involved in the prototyping**

Ans. 1. Define the goal and purpose of the prototyping.

2. Make plans for iterations (number, range) and evaluations (dates).

3. Transform the conceptual design to a first outline of the user interface and a first synopsis for the users' information.

4. Design the paper prototype.

5. Let domain experts review the paper prototype regarding completeness and correctness.

6. Test the prototype's usability.

7. Analyze the test results.

8. Decide on changes and additions regarding the conceptual model.

9. Design an executable prototype with users' information. Hold design meetings regularly, during which the following are documented:

- implemented changes and additions.

- unresolved requirements and problems.

10. Review and test the usability of the prototypes.

11. Analyze the test results.

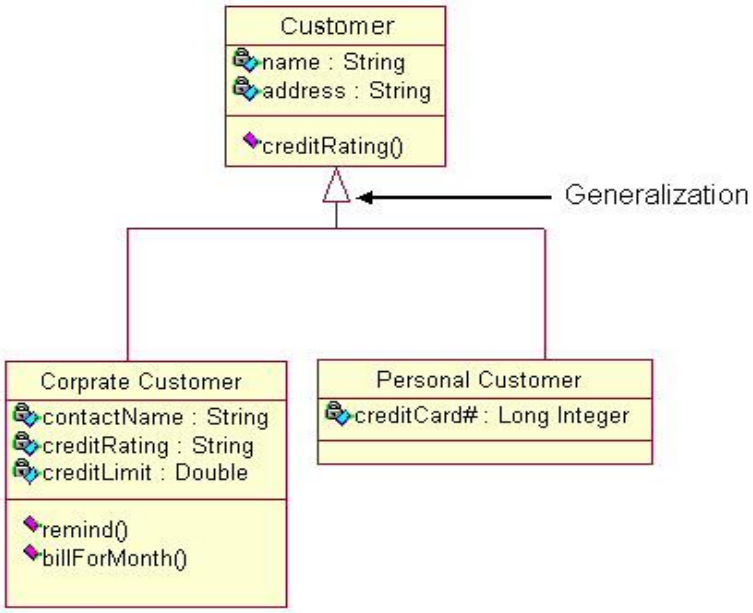
12. Decide on changes and additions regarding the conceptual model

13. Design the next prototype, and so on.

Q12. **What is generalization? Give an example of generalization.**

Ans. A generalization is an object class, which is a superset of another object class (or classes). Generalization models the "is a" relationship set since members of the specialization class (or classes) are always members of the generalization class. This means that members of the specialization class have all of the same properties of the generalization class including relationships with other objects as well as behaviour.

A generalization is used when two classes are similar, but have some differences. Look at the generalization below:



In this example the classes **Corporate Customer** and **Personal Customer** have some similarities such as name and address, but each class has some of its own attributes and operations. The class **Customer** is a general form of both the Corporate Customer and Personal Customer classes. This allows the designers to just use the Customer class for modules and do not require in-depth representation of each type of customer.

Q13. What are the Factors of Software Quality?

Ans. Factors of Software Quality:

• Portability-

• Usability

• Reusability

• Correctness

• Maintainability

Q14. Explain COCOMO model in detail.

Ans. Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size.

In COCOMO, projects are categorized into three types:

1. Organic
2. Semidetached
3. Embedded

1.Organic: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. **Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.**

2. Semidetached: A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. **Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management system.**

3. Embedded: A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. **For Example:** ATM, Air Traffic control.

For three product categories, Bohem provides a different set of expression to predict effort (in a unit of person month)and development time from the size of estimation in KLOC(Kilo Line

of code) efforts estimation takes into account the productivity loss due to holidays, weekly off, coffee breaks, etc.

According to Boehm, software cost estimation should be done through three stages:

1. Basic Model
2. Intermediate Model
3. Detailed Model

Q15. Differentiate between cohesion and coupling?

Coupling	Cohesion
Coupling is also called Inter-Module Binding.	Cohesion is also called Intra-Module Binding.
Coupling shows the relationships between modules.	Cohesion shows the relationship within the module.
Coupling shows the relative **independence** between the modules.	Cohesion shows the module's relative **functional** strength.
While creating, you should aim for low coupling, i.e., dependency among modules should be less.	While creating you should aim for high cohesion, i.e., a cohesive component/ module focuses on a single function (i.e., single-mindedness) with little interaction with other modules of the system.
In coupling, modules are linked to the other modules.	In cohesion, the module focuses on a single thing.

Q16. Explain the spiral model of software development with the help of a diagram. What are the limitations of this model?

Ans. The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

Spiral Model - Design

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

Identification

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

Design

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

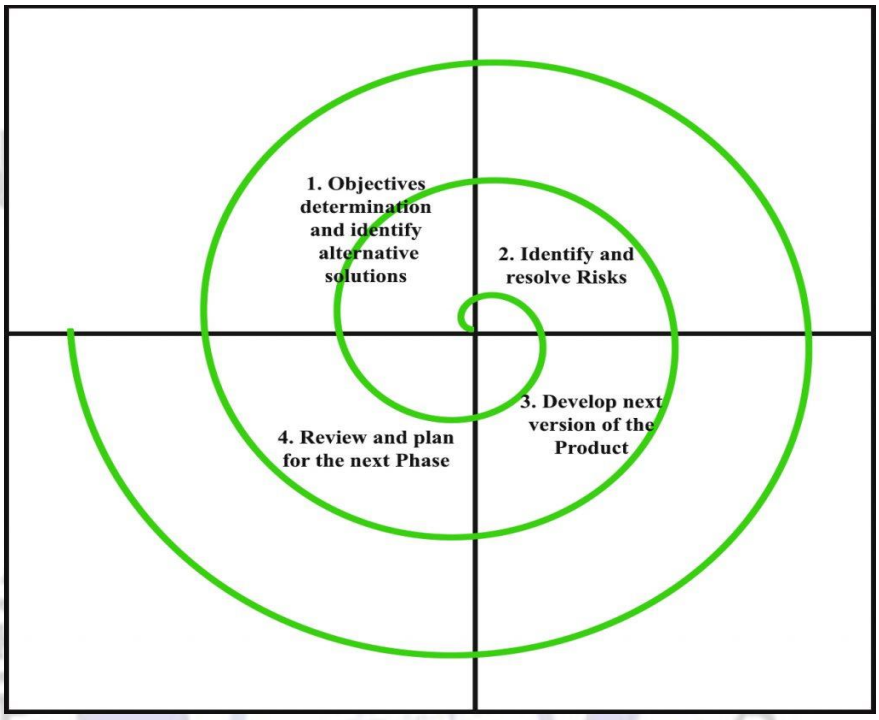
Construct or Build

The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

Evaluation and Risk Analysis

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.



Based on the customer evaluation, the software development process enters the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

The limitations of the Spiral SDLC Model are as follows −

- Management is more complex.
- End of the project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex
- Spiral may go on indefinitely.
- Large number of intermediate stages requires excessive documentation.

Q17. What is software maintenance? Also discuss the types of maintenance.

Ans. Software maintenance is widely accepted part of SDLC now a days. It stands for all the modifications and updations done after the delivery of software product. There are number of reasons, why modifications are required, some of them are briefly mentioned below:

- **Market Conditions** - Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger need for modification.
- **Client Requirements** - Over the time, customer may ask for new features or functions in the software.
- **Host Modifications** - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.
- **Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

Types of maintenance

In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance tasks as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:

- **Corrective Maintenance** - This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
- **Adaptive Maintenance** - This includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.
- **Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.

- **Preventive Maintenance** - This includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

Q18. What are the risk management activities?

Ans. **Risk management consists of three main activities:**

Risk Assessment

The objective of risk assessment is to division the risks in the condition of their loss, causing potential. For risk assessment, first, every risk should be rated in two methods:

- The possibility of a risk coming true (denoted as r).
- The consequence of the issues relates to that risk (denoted as s).

Based on these two methods, the priority of each risk can be estimated:

$$p = r * s$$

Where p is the priority with which the risk must be controlled, r is the probability of the risk becoming true, and s is the severity of loss caused due to the risk becoming true. If all identified risks are set up, then the most likely and damaging risks can be controlled first, and more comprehensive risk abatement methods can be designed for these risks.

1. Risk Identification: The project organizer needs to anticipate the risk in the project as early as possible so that the impact of risk can be reduced by making effective risk management planning.

A project can be of use by a large variety of risk. To identify the significant risk, this might affect a project. It is necessary to categories into the different risk of classes.

There are different types of risks which can affect a software project:

1. **Technology risks:** Risks that assume from the software or hardware technologies that are used to develop the system.
2. **People risks:** Risks that are connected with the person in the development team.
3. **Organizational risks:** Risks that assume from the organizational environment where the software is being developed.
4. **Tools risks:** Risks that assume from the software tools and other support software used to create the system.
5. **Requirement risks:** Risks that assume from the changes to the customer requirement and the process of managing the requirements change.
6. **Estimation risks:** Risks that assume from the management estimates of the resources required to build the system

2. Risk Analysis: During the risk analysis process, you have to consider every identified risk and make a perception of the probability and seriousness of that risk.

There is no simple way to do this. You have to rely on your perception and experience of previous projects and the problems that arise in them.

It is not possible to make an exact, the numerical estimate of the probability and seriousness of each risk. Instead, you should authorize the risk to one of several bands:

1. The probability of the risk might be determined as very low (0-10%), low (10-25%), moderate (25-50%), high (50-75%) or very high (+75%).
2. The effect of the risk might be determined as catastrophic (threaten the survival of the plan), serious (would cause significant delays), tolerable (delays are within allowed contingency), or insignificant.

Risk Control

It is the process of managing risks to achieve desired outcomes. After all, the identified risks of a plan are determined; the project must be made to include the most harmful and the most likely risks. Different risks need different containment methods. In fact, most risks need ingenuity on the part of the project manager in tackling the risk.

There are three main methods to plan for risk management:

1. **Avoid the risk:** This may take several ways such as discussing with the client to change the requirements to decrease the scope of the work, giving incentives to the engineers to avoid the risk of human resources turnover, etc.
2. **Transfer the risk:** This method involves getting the risky element developed by a third party, buying insurance cover, etc.
3. **Risk reduction:** This means planning method to include the loss due to risk. For instance, if there is a risk that some key personnel might leave, new recruitment can be planned.

Risk Leverage: To choose between the various methods of handling risk, the project plan must consider the amount of controlling the risk and the corresponding reduction of risk. For this, the risk leverage of the various risks can be estimated.

Risk leverage is the variation in risk exposure divided by the amount of reducing the risk.

Risk leverage = (risk exposure before reduction - risk exposure after reduction) / (cost of reduction)

1. Risk planning: The risk planning method considers each of the key risks that have been identified and develop ways to maintain these risks.

For each of the risks, you have to think of the behavior that you may take to minimize the disruption to the plan if the issue identified in the risk occurs.

You also should think about data that you might need to collect while monitoring the plan so that issues can be anticipated.

Again, there is no easy process that can be followed for contingency planning. It rely on the judgment and experience of the project manager.

2. Risk Monitoring: Risk monitoring is the method king that your assumption about the product, process, and business risks has not changed.

Q19. Describe the various strategies of design?

Ans. Software design is a process to conceptualize the software requirements into software implementation. Software design takes the user requirements as challenges and tries to find optimum solution. While the software is being conceptualized, a plan is chalked out to find the best possible design for implementing the intended solution.

There are multiple variants of software design.

Structured Design

Structured design is a conceptualization of problem into several well-organized elements of solution. It is basically concerned with the solution design. Benefit of structured design is, it gives better understanding of how the problem is being solved. Structured design also makes it simpler for designer to concentrate on the problem more accurately.

Structured design is mostly based on 'divide and conquer' strategy where a problem is broken into several small problems and each small problem is individually solved until the whole problem is solved.

The small pieces of problem are solved by means of solution modules. Structured design emphasis that these modules be well organized in order to achieve precise solution.

These modules are arranged in hierarchy. They communicate with each other. A good structured design always follows some rules for communication among multiple modules, namely -

Cohesion - grouping of all functionally related elements.

Coupling - communication between different modules.

A good structured design has high cohesion and low coupling arrangements.

Function Oriented Design

In function-oriented design, the system is comprised of many smaller sub-systems known as functions. These functions are capable of performing significant task in the system. The system is considered as top view of all functions.

Function oriented design inherits some properties of structured design where divide and conquer methodology is used.

This design mechanism divides the whole system into smaller functions, which provides means of abstraction by concealing the information and their operation.. These functional modules can share information among themselves by means of information passing and using information available globally.

Another characteristic of functions is that when a program calls a function, the function changes the state of the program, which sometimes is not acceptable by other modules. Function oriented design works well where the system state does not matter and program/functions work on input rather than on a state.

Design Process

- The whole system is seen as how data flows in the system by means of data flow diagram.
- DFD depicts how functions changes data and state of entire system.
- The entire system is logically broken down into smaller units known as functions on the basis of their operation in the system.
- Each function is then described at large.

Q20. Write short notes on: Reverse engineering and Re- engineering.

Ans. **Software Reverse Engineering** is a process of recovering the design, requirement specifications and functions of a product from an analysis of its code. It builds a program database and generates information from this.

The purpose of reverse engineering is to facilitate the maintenance work by improving the understand ability of a system and to produce the necessary documents for a legacy system.

Reverse Engineering Goals:

- Cope with Complexity.
- Recover lost information.

- Detect side effects.
- Synthesise higher abstraction.
- Facilitate Reuse.

Steps of Software Reverse Engineering:

1. **Collection Information:**
   This step focuses on collecting all possible information (i.e., source design documents etc.) about the software.
2. **Examining the information:**
   The information collected in step-1 as studied so as to get familiar with the system.
3. **Extracting the structure:**
   This step concerns with identification of program structure in the form of structure chart where each node corresponds to some routine.
4. **Recording the functionality:**
   During this step processing details of each module of the structure, charts are recorded using structured language like decision table, etc.
5. **Recording data flow:**
   From the information extracted in step-3 and step-4, set of data flow diagrams are derived to show the flow of data among the processes.
6. **Recording control flow:**
   High level control structure of the software is recorded.
7. **Review extracted design:**
   Design document extracted is reviewed several times to ensure consistency and correctness. It also ensures that the design represents the program.
8. **Generate documentation:**
   Finally, in this step, the complete documentation including SRS, design document, history, overview, etc. are recorded for future use.

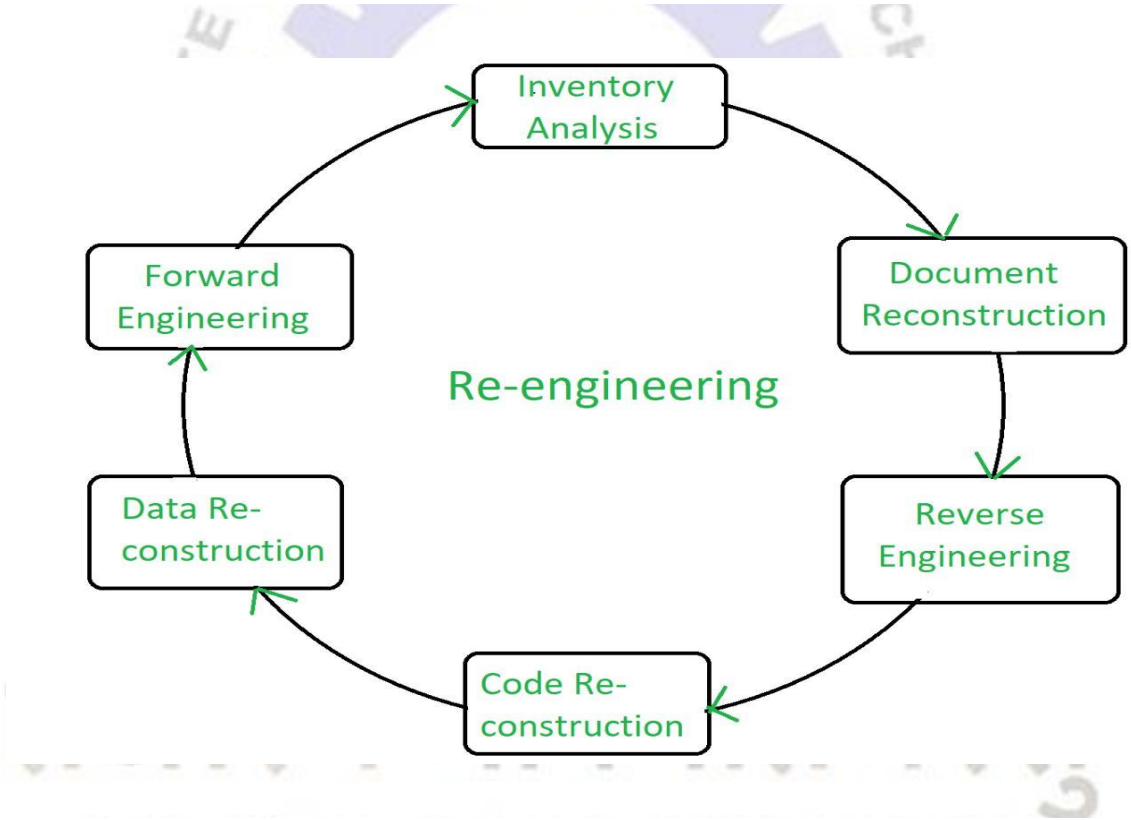
Software Re-engineering is a process of software development which is done to improve the maintainability of a software system. Re-engineering is the examination and alteration of a system to reconstitute it in a new form. This process encompasses a combination of sub-processes like reverse engineering, forward engineering, reconstructing etc.

Objectives of Re-engineering:

- To describe a cost-effective option for system evolution.
- To describe the activities involved in the software maintenance process.
- To distinguish between software and data re-engineering and to explain the problems of data re-engineering.

Steps involved in Re-engineering:

1. Inventory Analysis
2. Document Reconstruction
3. Reverse Engineering
4. Code Reconstruction
5. Data Reconstruction
6. Forward Engineering



Re-engineering Cost Factors:

- The quality of the software to be re-engineered
- The tool support available for re-engineering
- The extent of the required data conversion
- The availability of expert staff for re-engineering

Advantages of Re-engineering:

- **Reduced Risk:**

  As the software is already existing, the risk is less as compared to new software development. Development problems, staffing problems and specification problems are the lots of problems which may arise in new software development.

- **Reduced Cost:**

  The cost of re-engineering is less than the costs of developing new software.

BCA - IV Semester
Computer Networks
[BCA-210]

Q1. What is Data Communication? Explain its components in brief with the help of a diagram.

A1. Communication: To convey any message, data or thoughts from one place to another place using some medium is termed as a communication.

Components of data communication:

1. Sender
2. Message
3. Transmission Medium
4. Receiver
5. Protocols

Sender: Sender is the person who sends the message.

Message: Message is the information that is exchanged between sender and receiver.

Transmission Medium: Transmission Medium is the channel through which communication takes place.

Receiver: The person to whom the message is being sent is called 'receiver'.

Protocols: Protocols are set of rules which govern communication.

Q2. Explain the two types of line configuration: Point-to-Point configuration & Broadcasting/Multi-port configuration with the help of a diagram.

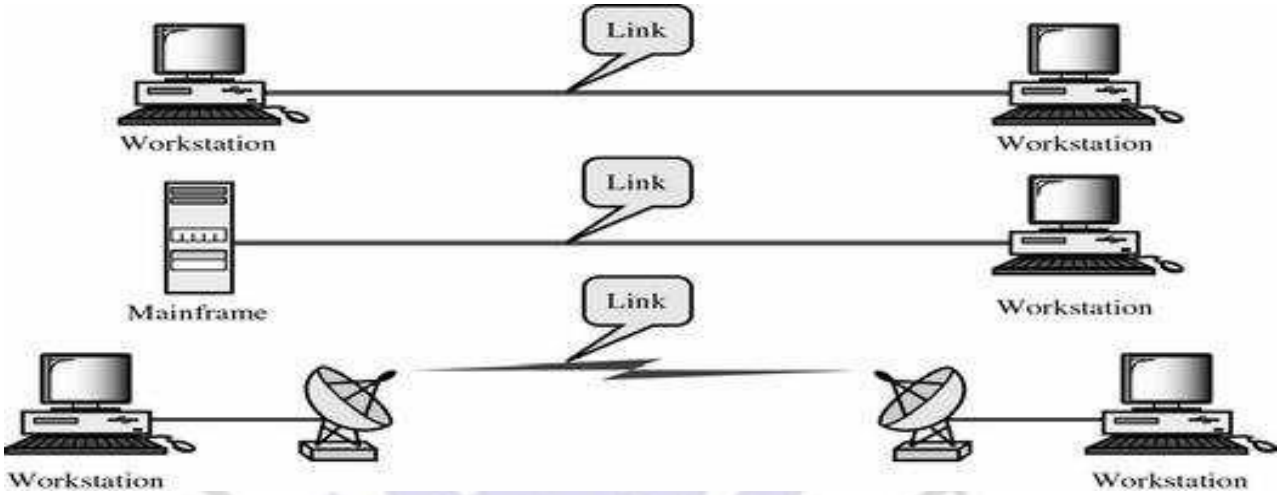
A2. Line configuration refers to the way two or more communication devices attached to a link. Line Configuration is also referred to as connection. A Link is the physical communication pathway that transfers data from one device to another. For communication to occur, two devices must be connected in same way to the same link at the same time.

Types of line configuration

1. Point-to-Point.
2. Multipoint.

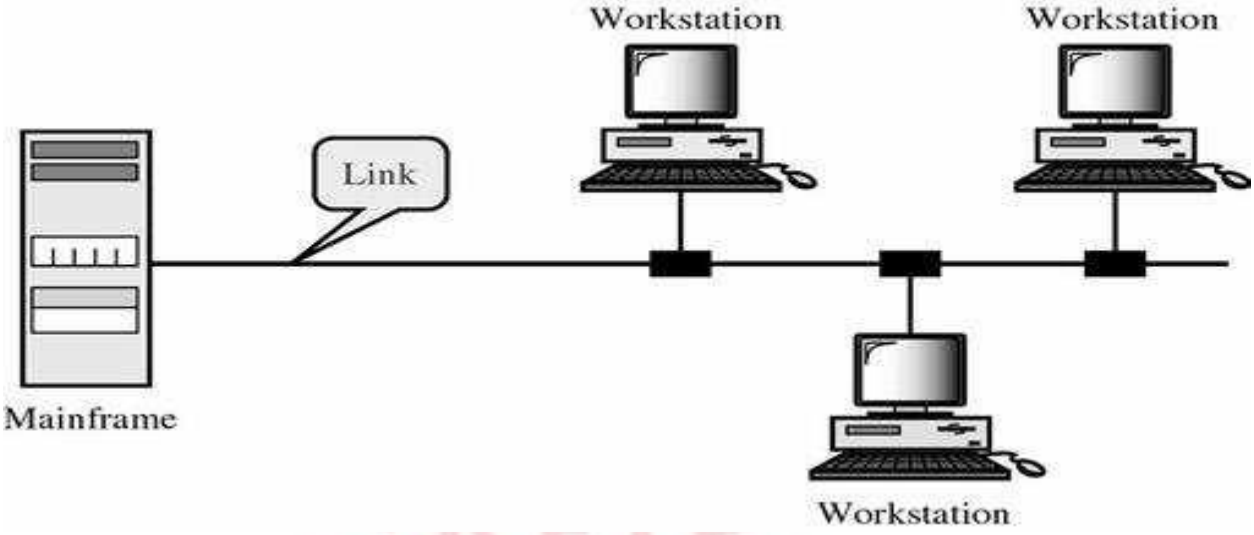
Point-to-Point:

A Point to Point Line Configuration Provide dedicated link between two devices use actual length of wire or cable to connect the two end including microwave & satellite link, Infrared remote control.



Multipoint Configuration:

Multipoint Configuration also known as Broadcasting, has one or more than two specific devices share a single link capacity of the channel is shared.

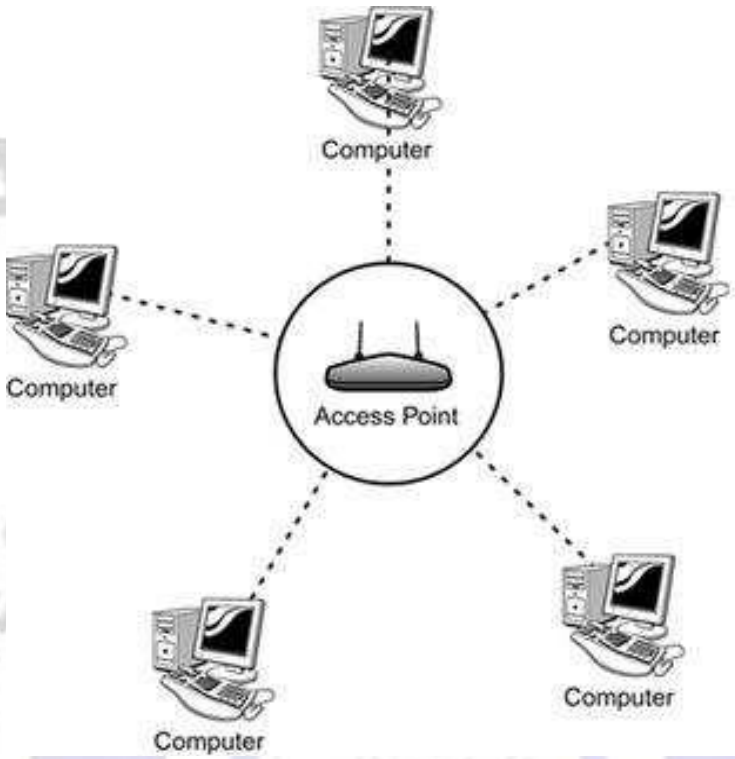


Q3. Define topology. Briefly explain star topology and bus topology, with the help of diagrams.

A3. The term "Topology" refers to the way in which the end points or stations/computer systems, attached to the networks, are interconnected. A topology is essentially a stable geometric arrangement of computers in a network.

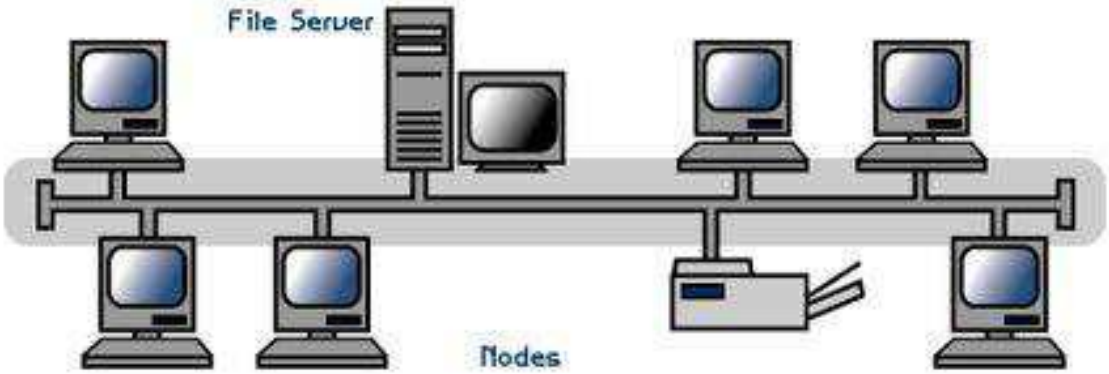
Star Topology:

In a star topology, cables run from every computer to a centrally located device called a HUB. Star topology networks require a central point of connection between media segment. These central points are referred to as Hubs. Hubs are special repeaters that overcome the electromechanical limitations of a media. Each computer on a star network communicates with a central hub that resends the message either to all the computers.



Bus topology:

A bus topology connects computers along a single or more cable to connect linearly. A network that uses a bus topology is referred to as a "bus network" which was the original form of Ethernet networks.



Q4. What do you understand by Transmission Mode?

A4. A given transmission on a communications channel between two machines can occur in several different ways.

Types of Transmission mode

- Simplex
- Half Duplex
- Full Duplex

A simplex connection is a connection in which the data flows in only one direction, from the transmitter to the receiver. This type of connection is useful if the data do not need to flow in both directions (for example, from your computer to the printer or from the mouse to your computer...).

A half-duplex connection (sometimes called an *alternating connection* or *semi-duplex*) is a connection in which the data flows in one direction or the other, but not both at the same time. With this type of connection, each end of the connection transmits in turn. This type of connection makes it possible to have bidirectional communications using the full capacity of the line.

A full-duplex connection is a connection in which the data flow in both directions simultaneously. Each end of the line can thus transmit and receive at the same time, which means that the bandwidth is divided in two for each direction of data transmission if the same transmission medium is used for both directions of transmission.

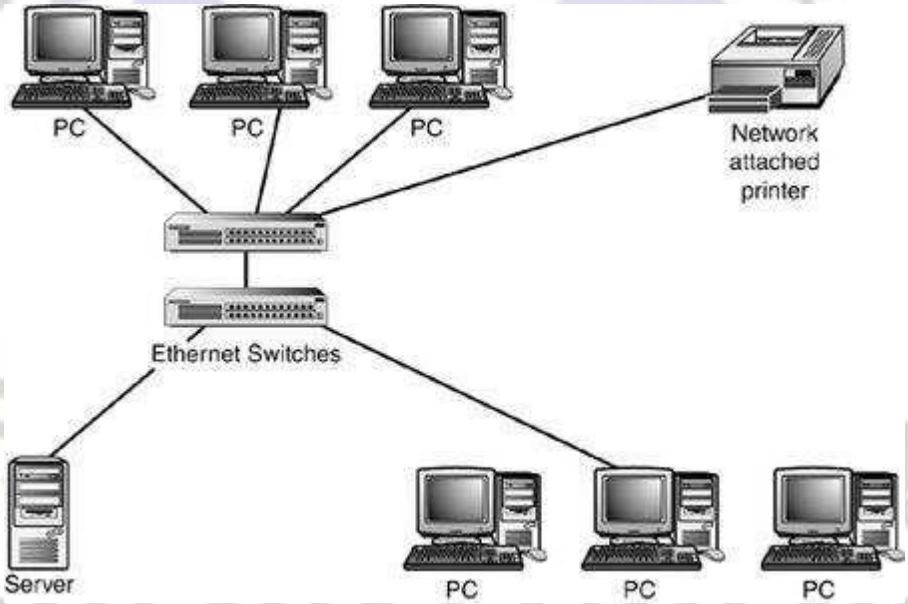
Q5. What are the different categories of networks available in networking?

A5. Categories of networks:

LAN - Local Area Network
MAN - Metropolitan Area Network
WAN - Wide Area Network

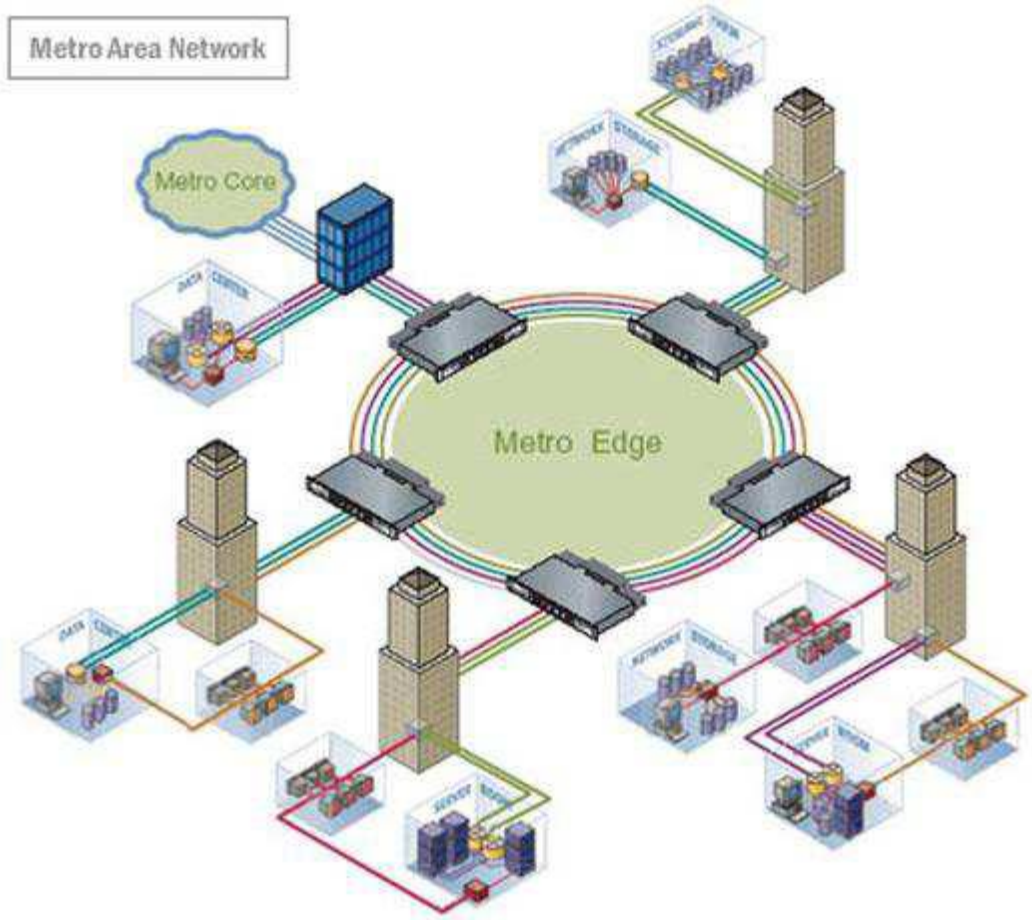
Local Area Network:

A LAN connects network devices over a relatively short distance. A networked office building, school, or home usually contains a single LAN, though sometimes one building will contain a few small LANs (perhaps one per room), and occasionally a LAN will span a group of nearby buildings. In addition to operating in a limited space, LANs are also typically owned, controlled, and managed by a single person or organization.



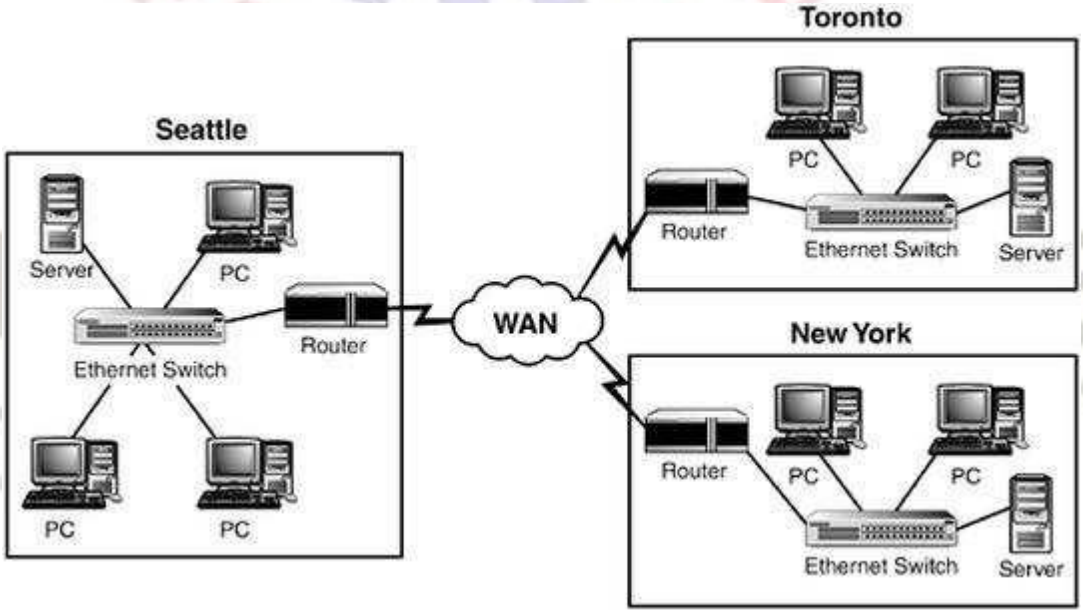
Metropolitan Area Network:

Any network spreading over a physical area larger than a LAN but smaller than a WAN, such as a city. A MAN is typically owned and operated by a single entity such as a government body or large corporation.



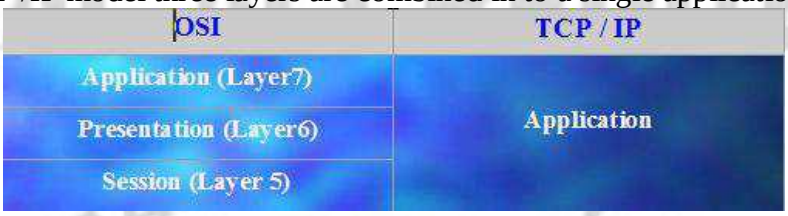
Wide Area Network:

A WAN is a network that spans more than one geographical location often connecting separated LANs. WANs are slower than LANs and often require additional and costly hardware such as routers, dedicated leased lines, and complicated implementation procedures.



Q6. How OSI reference model is different from TCP/IP model?
A6. OSI reference model is different from TCP/IP model in the following way:

- There are seven layers in OSI model where as TCP/IP has only five layers.
- In TCP /IP model three layers are combined in to a single application layer.



- The Session layer permits two parties to hold ongoing communications called a session across a network. Not found in TCP/IP model. In TCP/IP, its characteristics are provided by the TCP protocol.(Transport Layer)
- The Presentation Layer handles data format information for networked communications. This is done by converting data into a generic format that could be understood by both sides. Not found in TCP/IP model. In TCP/IP, this function is provided by the Application Layer. e.g. External Data Representation Standard (XDR) Multipurpose Internet Mail Extensions (MIME).
- The Application Layer is the top layer of the reference model. It provides a set of interfaces for applications to obtain access to networked services as well as access to the kinds of network services that support applications directly. OSI - FTAM, VT, MHS, DS, CMIP TCP/IP - FTP, SMTP, TELNET, DNS, SNMP Although the notion of an application process is common to both, their approaches to constructing application entities are different.
- Like all the other OSI Layers, the network layer provides both connectionless and connection-oriented services. As for the TCP/IP architecture, the internet layer is exclusively connectionless.
- Implementation of the OSI model places emphasis on providing a reliable data transfer service, while the TCP/IP model treats reliability as an end-to-end problem.
- Each layer of the OSI model detects and handles errors, all data transmitted includes checksums. The transport layer of the OSI model checks source-to-destination reliability.
- In the TCP/IP model, reliability control is concentrated at the transport layer. The transport layer handles all error detection and recovery. The TCP/IP transport layer uses checksums, acknowledgments, and timeouts to control transmissions and provides end-to- end verification.
- Hosts on OSI implementations do not handle network operations (simple terminal), but TCP/IP hosts participate in most network protocols.
- TCP/IP hosts carry out such functions as end-to-end verification, routing, and network control. The TCP/IP internet can be viewed as a data stream delivery system involving intelligent hosts.

Q7. Explain Transmission Media in detail.
A7. Transmission media means any medium used for communication. It can be divided into two categories':

1. Guided media
2. Unguided media

Guide media is that where we use any path for communication like cables (coaxial, fibre optic, twisted pair) etc. Examples of guided media are: - Twisted Pair Cable, Co-axial Cable, Optical Fiber Cable.

Unguided media is also called wireless where not any physical path is used for transmission. Examples of unguided media are: - Microwave or Radio Links, Infrared.

There are three categories of guided media:
Twisted-pair cable
Coaxial cable
Fiber-optic cable

Twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. Twisted-pair cable comes in two forms: unshielded and shielded

The twisting helps to reduce the interference (noise) and crosstalk.

Figure 7.4 *Frequency range for twisted-pair cable*

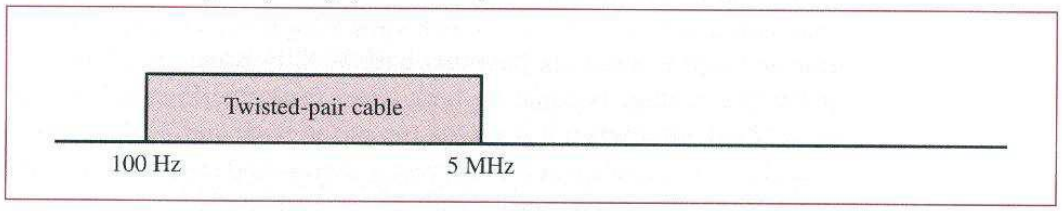
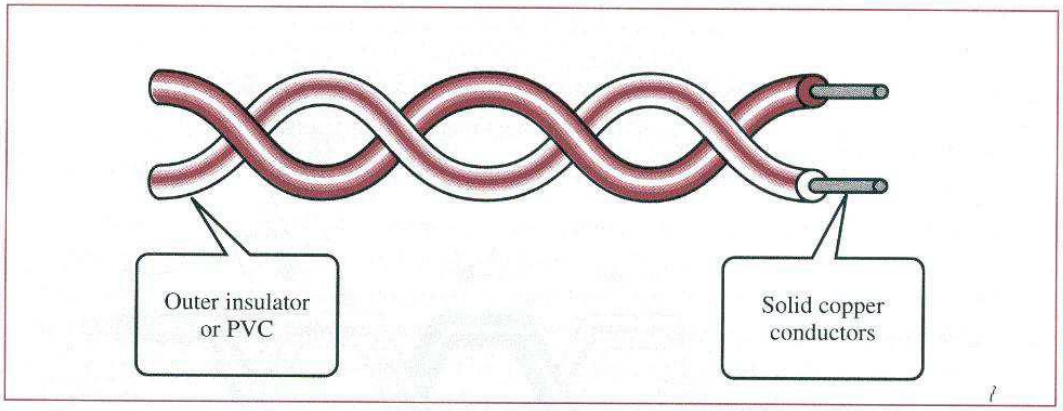


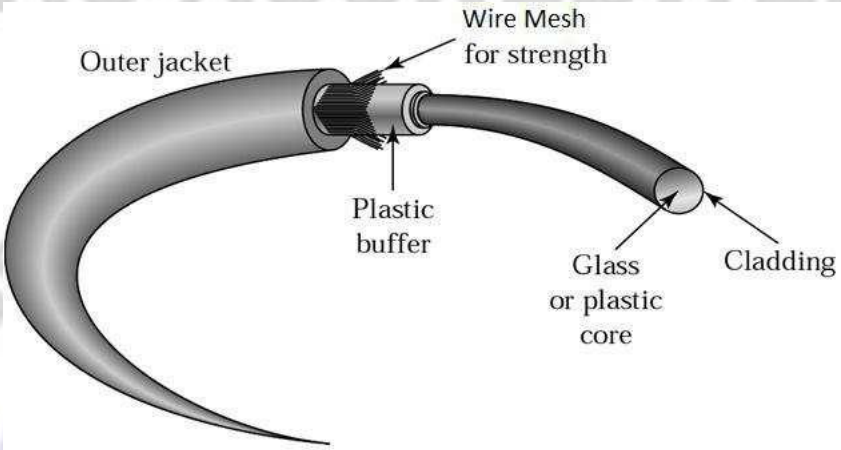
Figure 7.5 *Twisted-pair cable*



Coaxial cable carries signals of higher frequency ranges than twisted-pair cable. It has inner conductor, Insulator, Outer conductor metal mesh, Insulator and plastic cover.
Applications:
. Television distribution
. Cable TV
. Long distance telephone transmission
. Can carry 10,000 voice calls simultaneously
. Short distance computer systems links
. Local area networks
. More expensive than twisted pair, not as popular for LANs

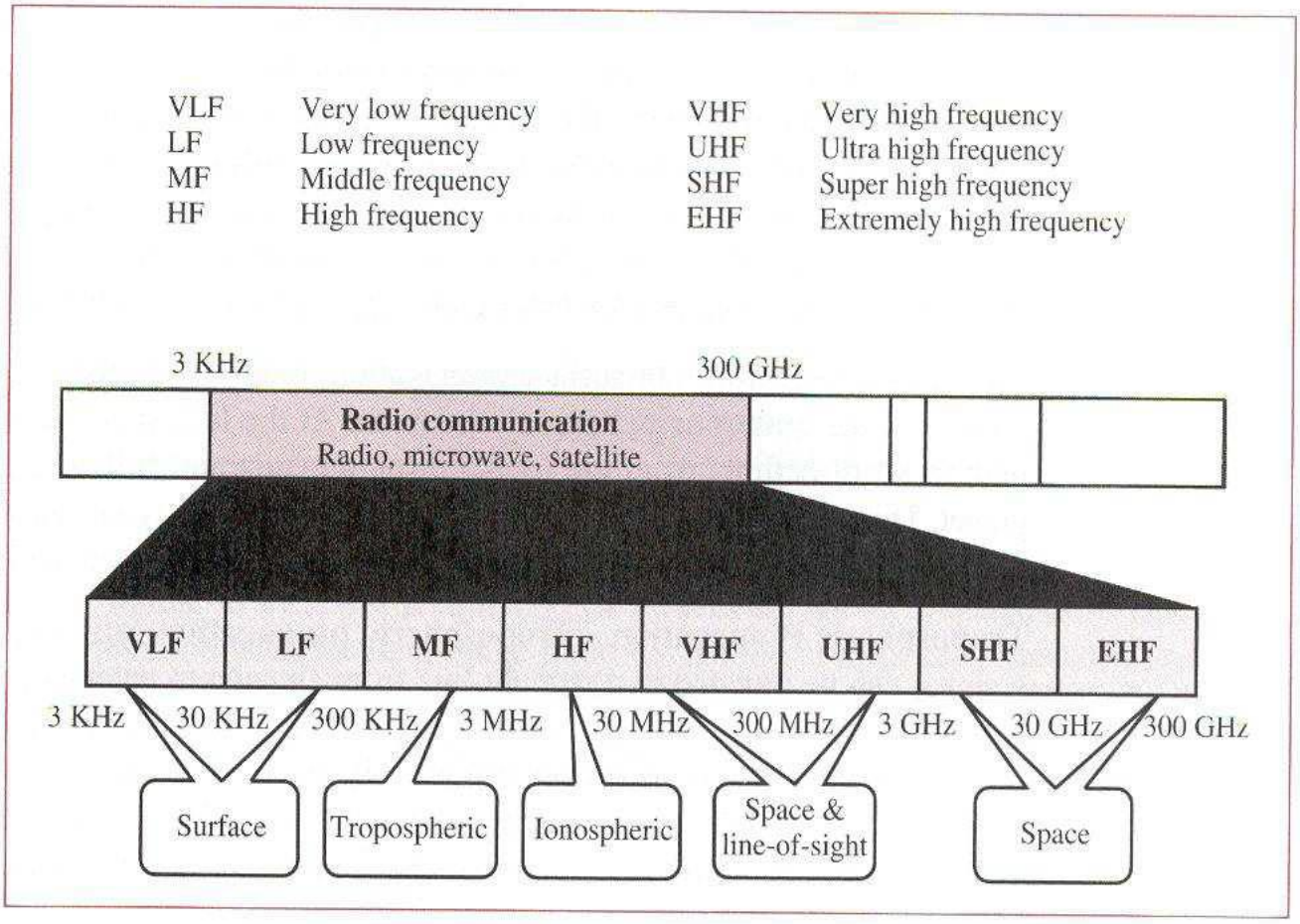
Fiber optics cable:
. Metal cables transmit signals in the form of electric current.

. Optical fiber is made of glass or plastic and transmits signals in the form of light.
. Light, a form of electromagnetic energy, travels at 300,000 Kilometers/second (186,000 miles/second), in a vacuum.
. The speed of the light depends on the density of the medium through which it is traveling (the higher density, the slower the speed).



Unguided media is also called wireless where not any physical path is used for transmission. Examples of unguided media are: - Microwave or Radio Links, Infrared.

Figure 7.21 *Radio communication band*



Q8. What do you understand by Transmission Impairment? Explain.

A8. Transmission impairments:

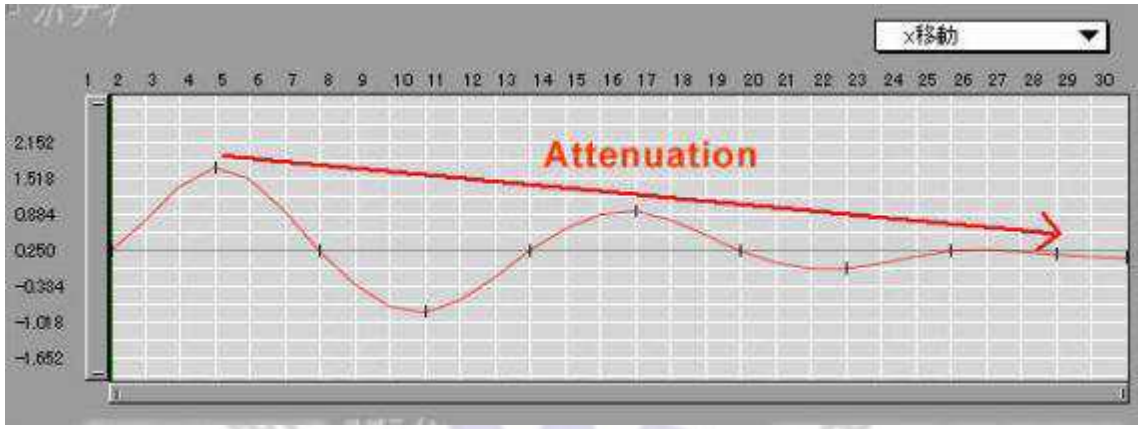
1. Attenuation
2. Distortion
3. Noise

Attenuation: In computer networking, attenuation is a loss of signal strength measured in decibels (dB). Attenuation occurs on networks for several reasons:

Range - both wireless and wired transmissions gradually dissipate in strength over longer reaches.

Interference - on wireless networks, radio interference or physical obstructions like walls also dampen communication signals.

Wire size - on wired networks, thinner wires suffer from higher (more) attenuation than thicker wires.



Distortion:

Various frequency components making up the signal arrive at the receiver with varying delays.

Inter symbol Interference - the frequency components are delayed and they start to interfere with the frequency components associated with the later bit.

Only in guided media.

Propagation velocity varies with frequency.

Noise:

Thermal Agitates the electrons in conductors, and is a function of the temperature. It is often referred to as

white noise, because it affects uniformly the different frequencies.

Inter modulation Resulting from interference of different frequencies sharing the same medium. It is caused

by a component malfunction or a signal with excessive strength is used.

Crosstalk Foreign signal enters the path of the transmitted signal.

Impulse Irregular disturbances, such as lightning, and flawed communication elements. It is a primary source

of error in digital data.

Q9. Write short note on FDM & TDM.

A9. Frequency Division Multiplexing (FDM):

It is the basis for broadcast radio. Several stations can transmit simultaneously without interfering with each other provided they use separate carrier frequencies (separate channels). In data communications FDM is implemented by sending multiple carrier waves over the same copper wire. At the receiver's end, de-multiplexing is performed by filtering out the frequencies other than the one carrying the expected transmission. Any of the modulation methods discussed before can be used to carry bits within a channel.

Time Division Multiplexing (TDM):

It means dividing the available transmission time into time slots, and allocating a different slot to each transmitter. One method for transmitters to take turns is to transmit in round-robin order.

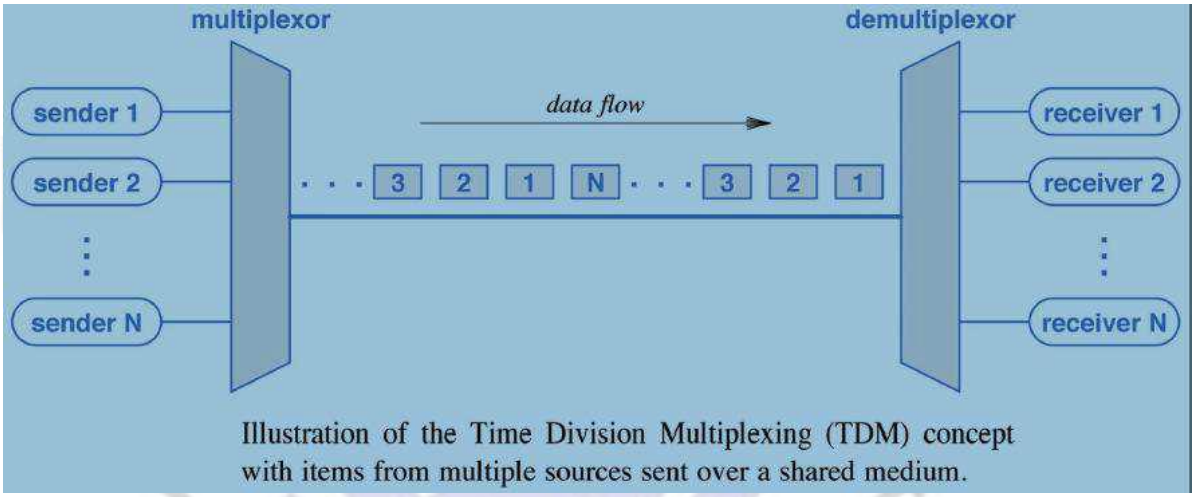
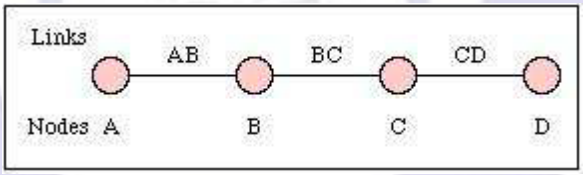


Illustration of the Time Division Multiplexing (TDM) concept with items from multiple sources sent over a shared medium.

Q10. Describe various types of switching techniques.

A10. Circuit switching:

Circuit switching is the most familiar technique used to build a communications network. It is used for ordinary telephone calls. It allows communications equipment and circuits, to be shared among users. Each user has sole access to a circuit (functionally equivalent to a pair of copper wires) during network use. Consider communication between two points A and D in a network. The connection between A and D is provided using (shared) links between two other pieces of equipment, B and C.

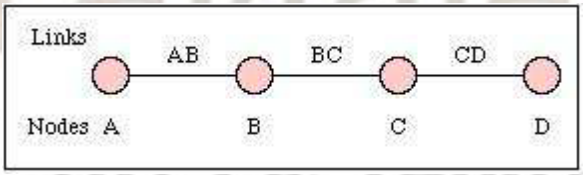


A connection between two systems A & D formed from 3 links

Packet switching:

Packet switching is similar to message switching using short messages. Any message exceeding a network-defined maximum length is broken up into shorter units, known as packets, for transmission; the packets, each with an associated header, are then transmitted individually through the network. The fundamental difference in packet communication is that the data is formed into packets with a pre-defined header format (i.e. PCI), and well-known "idle" patterns which are used to occupy the link when there is no data to be communicated.

Packet network equipment discards the "idle" patterns between packets and processes the entire packet as one piece of data. The equipment examines the packet header information (PCI) and then either removes the header (in an end system) or forwards the packet to another system. If the out-going link is not available, then the packet is placed in a queue until the link becomes free. A packet network is formed by links which connect packet network equipment.



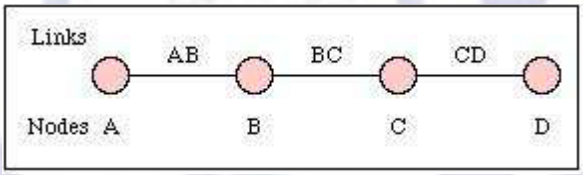
Communication between A and D using circuits which are shared using packet switching.

There are two important benefits from packet switching.

1. The first and most important benefit is that since packets are short, the communication links between the nodes are only allocated to transferring a single message for a short period of time while transmitting each packet. Longer messages require a series of packets to be sent, but do not require the link to be dedicated between the transmission of each packet. The implication is that packets belonging to other messages may be sent between the packets of the message being sent from A to D. This provides a much fairer sharing of the resources of each of the links.

2. Another benefit of packet switching is known as "pipelining". Pipelining is visible in the figure above. At the time packet 1 is sent from B to C, packet 2 is sent from A to B; packet 1 is sent from C to D while packet 2 is sent from B to C, and packet 3 is sent from A to B, and so forth. This simultaneous use of communications links represents a gain in efficiency; the total delay for transmission across a packet network may be considerably less than for message switching, despite the inclusion of a header in each packet rather than in each message.

Message switching:
Sometimes there is no need for a circuit to be established all the way from the source to the destination. Consider a connection between the users (A and D) in the figure below (i.e. A and D) is represented by a series of links (AB, BC, and CD).



A connection between two systems A & D formed from 3 links

For instance, when a telex (or email) message is sent from A to D, it first passes over a local connection (AB). It is then passed at some later time to C (via link BC), and from there to the destination (via link CD). At each message switch, the received message is stored, and a connection is subsequently made to deliver the message to the neighboring message switch. Message switching is also known as store-and-forward switching since the messages are stored at intermediate nodes en route to their destinations.
The figure illustrates message switching; transmission of only one message is illustrated for simplicity. As the figure indicates, a complete message is sent from node A to node B when the link interconnecting them becomes available. Since the message may be competing with other messages for access to facilities, a queuing delay may be incurred while waiting for the link to become available. The message is stored at B until the next link becomes available, with another queuing delay before it can be forwarded. It repeats this process until it reaches its destination.

Q11. What are ISDN, Subscribers' Access and B-ISDN?
A11. ISDN:
Integrated Services Digital Network (ISDN) is a set of communication standards for digital transmission of voice, video, data, and other network services over the traditional circuits of the public switched telephone network.

Subscribers' Access:
ISDN is a circuit-switched telephone network system, which also provides access to packet switched networks, designed to allow digital transmission of voice and data over ordinary telephone copper wires,

resulting in potentially better voice quality than an analog phone can provide. It offers circuit-switched connections (for either voice or data), and packet-switched connections (for data), in increments of 64 kilobit/s. A major market application for ISDN in some countries is Internet access, where ISDN typically provides a maximum of 128 kbps in both upstream and downstream directions. Channel bonding can achieve a greater data rate; typically the ISDN B-channels of three or four BRIs (six to eight 64 kbps channels) are bonded.

Broadband ISDN: Broadband Integrated Services Digital Network (BISDN)

Broadband Integrated Services Digital Network (BISDN or Broadband ISDN) is designed to handle high bandwidth applications. BISDN currently uses ATM technology over SONET-based transmission circuits to provide data rates from 155 to 622Mbps and beyond, contrast with the traditional narrowband ISDN (or N-ISDN), which is only 64 Kbps basically and up to 2 Mbps. The designed Broadband ISDN (BISDN) services can be categorized as follows:

. Conversational services such as telephone-like services, which was also supported by N-ISDN.

Also the additional bandwidth offered will allow such services as video telephony, video conferencing and high volume, high speed data transfer.

. Messaging services, which is mainly a store-and-forward type of service. Applications could include voice and video mail, as well as multi-media mail and traditional electronic mail.

. Retrieval services which provides access to (public) information stores, and information is sent to the user on demand only.

. No user control of presentation. This would be for instance, a TV broadcast, where the user can choose simply either to view or not.

. User controlled presentation. This would apply to broadcast information that the user can partially control.

Q12. Write short notes on Bridges & Gateways.

A12. Bridges: A network bridge connects multiple network segments at the data link layer of the OSI model. Bridges do not promiscuously copy traffic to all ports, as a hub do, but learns which MAC addresses are reachable through specific ports. Once the bridge associates a port and an address, it will send traffic for that address only to that port. Bridges do send broadcasts to all ports except the one on which the broadcast was received. Bridges learn

the association of ports and addresses by examining the source address of frames that it sees on various ports. Once a frame arrives through a port, its source address is stored and the bridge assumes that MAC address is associated with that port. The first time that a previously unknown destination address is seen, the bridge will forward the frame to all ports other than the one on which the frame arrived.

Gateways: Gateways work on all seven OSI layers. The main job of a gateway is to convert protocols among communications networks. A router by itself transfers, accepts and relays packets only across networks using similar protocols. A gateway on the other hand can accept a packet formatted for one protocol (e.g. AppleTalk) and convert it to a packet formatted for another protocol (e.g. TCP/IP) before forwarding it. A gateway can be implemented in hardware, software or both, but they are usually implemented by software installed within a router. A gateway must understand the protocols used by each network linked into the router. Gateways are slower than bridges, switches and (non-gateway) routers. A gateway is a network point that acts as an entrance to another network. On the Internet, a node or stopping point can be either a gateway node or a host (end-point) node. Both the computers of Internet users and the computers that serve pages to users are host nodes, while the nodes that connect the networks in between are gateways. For example, the computers that control traffic between company networks or the computers used by internet service providers (ISPs) to connect users to the internet are gateway nodes.

Q13. What is Routing? How Static Routing is different from Dynamic Routing?
A13. Routing: is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered.

Static routing algorithms are hardly algorithms at all, but are table mappings established by the network administrator before the beginning of routing. These mappings do not change unless the network administrator alters them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple. Because static routing systems cannot react to network changes, they generally are considered unsuitable for today's large, constantly changing networks. Most of the dominant routing algorithms today are dynamic routing algorithms, which adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly.

Dynamic routing algorithms can be supplemented with static routes where appropriate. A router of last resort (a router to which all unroutable packets are sent), for example, can be designated to act as a repository for all unroutable packets, ensuring that all messages are at least handled in some way.

Q14. Describe Hierarchical Routing.
A14. In a hierarchical routing system, some routers form what amounts to a routing backbone. Packets from non backbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination. At this point, they travel from the last backbone router through one or more non backbone routers to the final destination. Routing systems often designate logical groups of nodes, called domains, autonomous systems, or areas. In hierarchical systems, some routers in a domain can communicate with routers in other domains, while others can communicate only with routers within their domain. In very large networks, additional hierarchical levels may exist, with routers at the highest hierarchical level forming the routing backbone. The primary advantage of hierarchical routing is that it mimics the organization of most companies and therefore supports their traffic patterns well. Most network communication occurs within small company groups (domains). Because intra-domain routers need to know only about other routers within their domain, their routing algorithms can be simplified, and, depending on the routing algorithm being used, routing update traffic can be reduced accordingly.

Q15. Explain Distance Vector and Link State Routing.
A15. Link-state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the internetwork. Each router, however, sends only the portion of the routing table that describes the state of its own links. In link-state algorithms, each router builds a picture of the entire network in its routing tables. Distance vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers.
Distance vector algorithms know only about their neighbors. Because they converge more quickly, link-state algorithms are somewhat less prone to routing loops than distance vector algorithms. On the other hand, link-state algorithms require more CPU power and memory than distance vector algorithms. Link-state algorithms, therefore, can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

Q16. Briefly explain Transport Layer of OSI Model and its protocols.

A16. In computer networking, the **transport layer** or **layer 4** provides end-to-end communication services for applications within a layered architecture of network components and protocols. The transport layer provides convenient services such as connection-oriented data stream support, reliability and flow control.

Connection-oriented communication: It is normally easier for an application to interpret a connection as a data stream rather than having to deal with the underlying connection-less models, such as the datagram model of the User Datagram Protocol (UDP) and of the Internet Protocol (IP).

Byte orientation: Rather than processing the messages in the underlying communication system format, it is often easier for an application to process the data stream as a sequence of bytes. This simplification helps applications work with various underlying message formats.

Same order delivery: The network layer doesn't generally guarantee that packets of data will arrive in the same order that they were sent, but often this is a desirable feature. This is usually done through the use of segment numbering, with the receiver passing them to the application in order. This can cause head-of-line blocking.

Reliability: Packets may be lost during transport due to network congestion and errors. By means of an error detection code, such as a checksum, the transport protocol may check that the data is not corrupted, and verify correct receipt by sending an ACK or NACK message to the sender. Automatic repeat request schemes may be used to retransmit lost or corrupted data.

Flow control: The rate of data transmission between two nodes must sometimes be managed to prevent a fast sender from transmitting more data than can be supported by the receiving data buffer, causing a buffer overrun. This can also be used to improve efficiency by reducing buffer under run.

Congestion avoidance: Congestion control can control traffic entry into a telecommunications network, so as to avoid congestive collapse by attempting to avoid oversubscription of any of the processing or link capabilities of the intermediate nodes and networks and taking resource reducing steps, such as reducing the rate of sending packets.

Different protocols used in Transport layer

TCP

UDP

The most well-known transport protocol is the Transmission Control Protocol (TCP). It lent its name to the title of the entire Internet Protocol Suite, *TCP/IP*. It is used for connection-oriented transmissions, whereas the connectionless User Datagram Protocol (UDP) is used for simpler messaging transmissions. TCP is the more complex protocol, due to its stateful design incorporating reliable transmission and data stream services.

There are many services that can be optionally provided by a transport-layer protocol, and different protocols may or may not implement them.

Q17. What do you understand by Connection Management?

A17. Connection management:

A symmetric connection management service between two service access points is specified, using a state transition system and safety and progress requirements. At each access point. the user can request connection establishment, request connection termination, and signal whether or not they are willing to accept connection requests from the remote user. The protocol can indicate connection establishment, connection termination, and rejection of a connection establishment request. The authors then specify a protocol and verify that it offers the service, given communication channels between the access points that can lose, reorder, and duplicate messages, but which guarantee delivery of a message that is repeatedly sent. The protocol achieves the service using 2-way and 3-way handshakes, and can be directly combined with any existing single-connection data transfer protocols to provide a transport layer protocol that offers both connection management and data transfer services.

Three way hand shaking:

Before the sending device and the receiving device start the exchange of data, both devices need to be synchronized. During the TCP initialization process, the sending device and the receiving device exchange a few control packets for synchronization purposes. This exchange is known as a three-way handshake. The three-way handshake begins with the initiator sending a TCP segment with the SYN control bit flag set. TCP allows one side to establish a connection. The other side may either accept the connection or refuse it. If we consider this from application layer point of view, the side that is establishing the connection is the client and the side waiting for a connection is the server.

TCP identifies two types of OPEN calls:

Active Open: In an Active Open call a device (client process) using TCP takes the active role and initiates the connection by sending a TCP SYN message to start the connection.

Passive Open: A passive OPEN can specify that the device (server process) is waiting for an active OPEN from a specific client. It does not generate any TCP message segment. The server processes listening for the clients are in Passive Open mode.

Q18. What are the functions of Session Layer and Presentation Layer?

A18. Session layers:

The session protocol defines the format of the data sent over the connections. Session layer establish and manages the session between the two users at different ends in a network. Session layer also manages who can transfer the data in a certain amount of time and for how long. The examples of session layers and the interactive logins and file transfer sessions. Session layer reconnect the session if it disconnects. It also reports and logs and upper layer errors.

The session layer allows session establishment between processes running on different stations.

Functions of Session layer:

. **Session establishment, maintenance and termination**: allows two application processes on different machines to establish, use and terminate a connection, called a session.

. **Session support**: performs the functions that allow these processes to communicate over the network, performing security, name recognition, logging and so on.

. **Protocols:** The protocols that work on the session layer are NetBIOS, Mail Slots, Names Pipes, RPC.

Presentation layer:

Presentation layer is also called translation layer. The presentation layer presents the data into a uniform format and masks the difference of data format between two dissimilar systems The presentation layer formats the data to be presented to the application layer. It can be viewed as the translator for the network. This layer may translate data from a format used by the application layer into a common format at the sending station, and then translate the common format to a format known to the application layer at the receiving station.

Functions of Presentation layer:

. Character code translation: for example, ASCII to EBCDIC.

. Data conversion: bit order, CR-CR/LF, integer-floating point, and so on.

. Data compression: reduces the number of bits that need to be transmitted on the network.

. Data encryption: encrypt data for security purposes. For eg., password encryption.

Q19. Explain Hamming Code with the help of an example.

A19. Suppose a binary data 1001101 is to be transmitted. To implement hamming code for this, following steps are used:

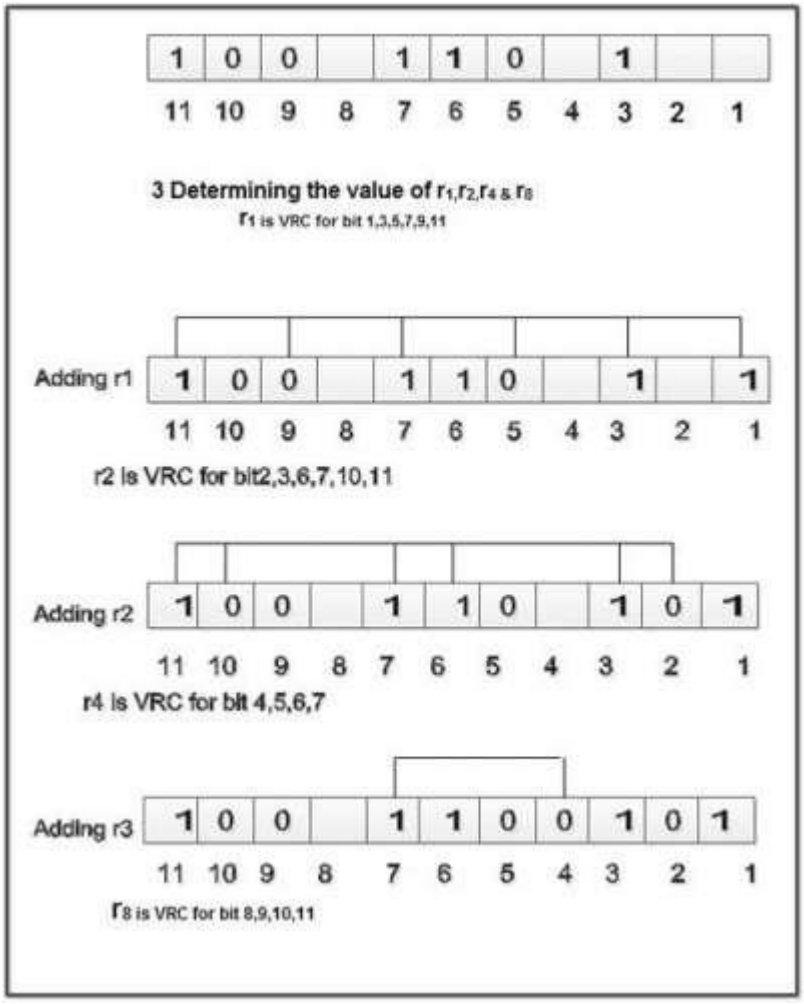
1. Calculating the number of redundancy bits required. Since number of data bits is 7, the value of r is calculated as

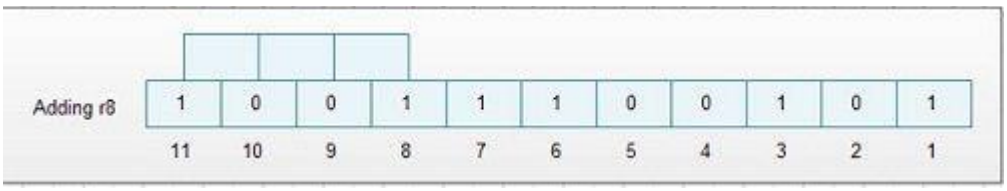
$$2^r \geq m + r + 1$$

$$2^4 \geq 7 + 4 + 1$$

Therefore no. of redundancy bits = 4

2. Determining the positions of various data bits and redundancy bits. The various r bits are placed at the position that corresponds to the power of 2 *i.e.* 1, 2, 4, 8

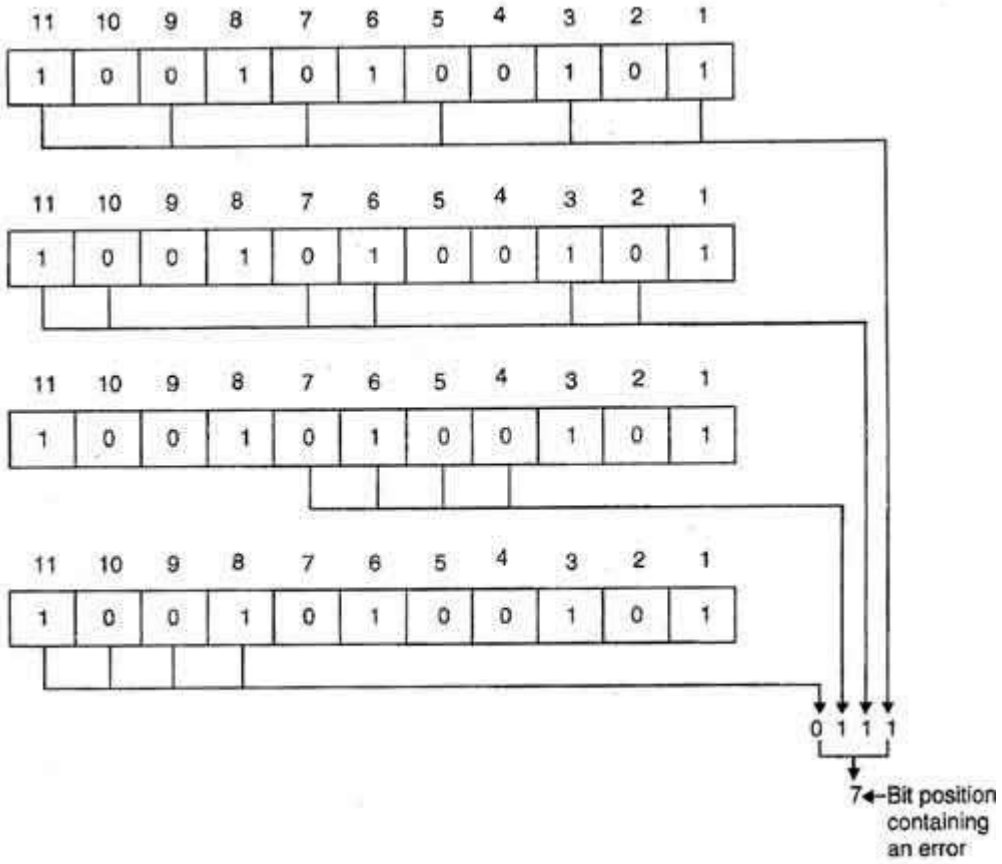




4. Thus data 1 0 0 1 1 1 0 0 1 0 1 with be transmitted.

Error Detection & Correction

Considering a case of above discussed example, if bit number 7 has been changed from 1 to 0.The data will be erroneous.



Data sent: 1 0 0 1 1 1 0 0 1 0 1

Data received: 1 00 1 0 1 00 1 0 1 (seventh bit changed)

The receive takes the transmission and recalculates four new VRCs using the same set of bits used by sender plus the relevant parity (r) bit for each set as shown in fig.

Then it assembles the new parity values into a binary number in order of r position (r_8, r_4, r_2, r_1).

In this example, this step gives us the binary number 0111. This corresponds to decimal 7. Therefore bit number 7 contains an error. To correct this error, bit 7 is reversed from 0 to 1.

Q20. Explain Leaky Bucket and Token Bucket Algorithms.

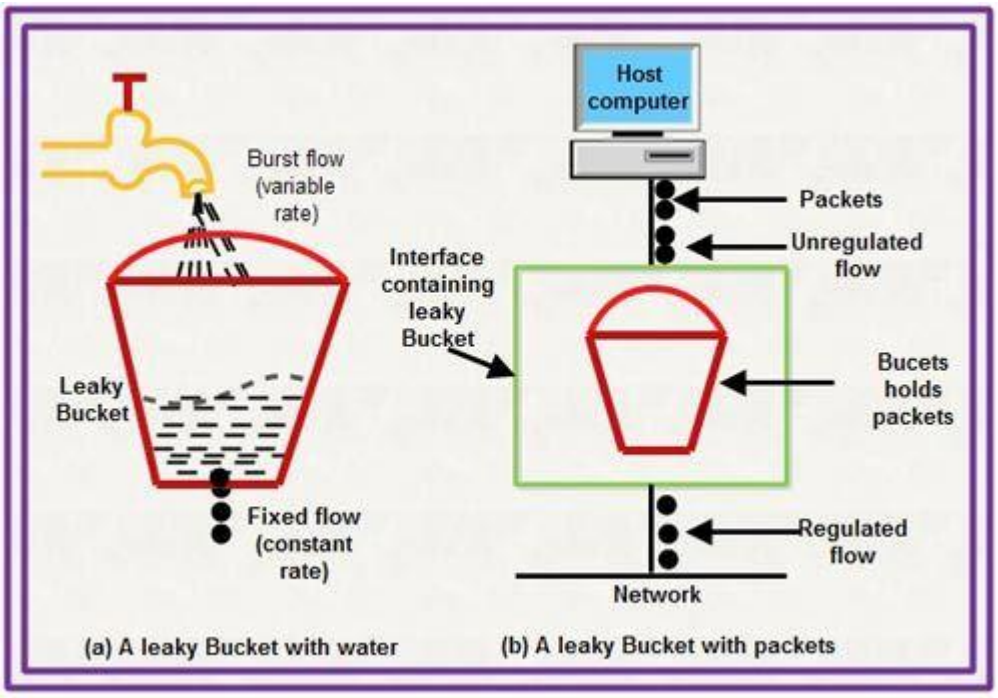
A20. Leaky Bucket Algorithm

• It is a traffic shaping mechanism that controls the amount and the rate of the traffic sent to the network.

• A leaky bucket algorithm shapes bursty traffic into fixed rate traffic by averaging the data rate.

• Imagine a bucket with a small hole at the bottom.

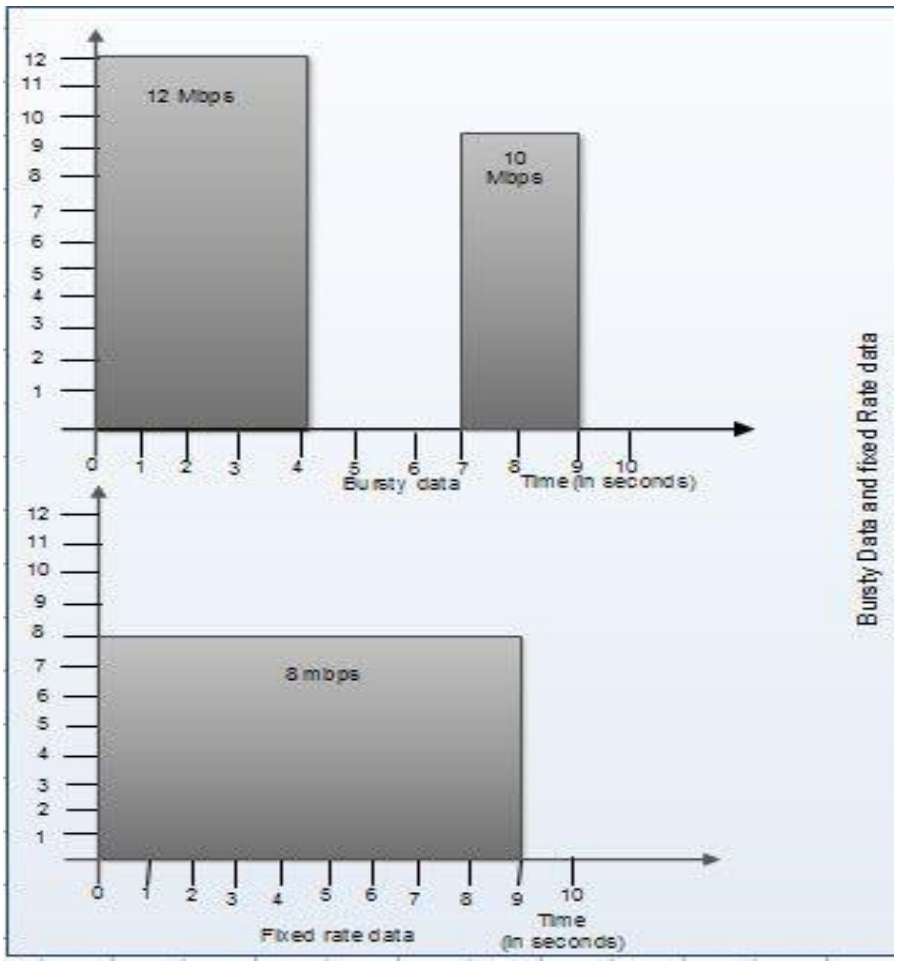
• The rate at which the water is poured into the bucket is not fixed and can vary but it leaks from the bucket at a constant rate. Thus (as long as water is present in bucket), the rate at which the water leaks does not depend on the rate at which the water is input to the bucket.



• Also, when the bucket is full, any additional water that enters into the bucket spills over the sides and is lost.

• The same concept can be applied to packets in the network. Consider that data is coming from the source at variable speeds. Suppose that a source sends data at 12 Mbps for 4 seconds. Then there is no data for 3 seconds. The source again transmits data at a rate of 10 Mbps for 2 seconds. Thus, in a time span of 9 seconds, 68 Mb data has been transmitted.

If a leaky bucket algorithm is used, the data flow will be 8 Mbps for 9 seconds. Thus constant flow is maintained.



Token bucket Algorithm

• The leaky bucket algorithm allows only an average (constant) rate of data flow. Its major problem is that it cannot deal with bursty data.

• A leaky bucket algorithm does not consider the idle time of the host. For example, if the host was idle for 10 seconds and now it is willing to sent data at a very high speed for another 10 seconds, the total data transmission will be divided into 20 seconds and average data rate will be maintained. The host is having no advantage of sitting idle for 10 seconds.

• To overcome this problem, a token bucket algorithm is used. A token bucket algorithm allows bursty data transfers.

• A token bucket algorithm is a modification of leaky bucket in which leaky bucket contains tokens.

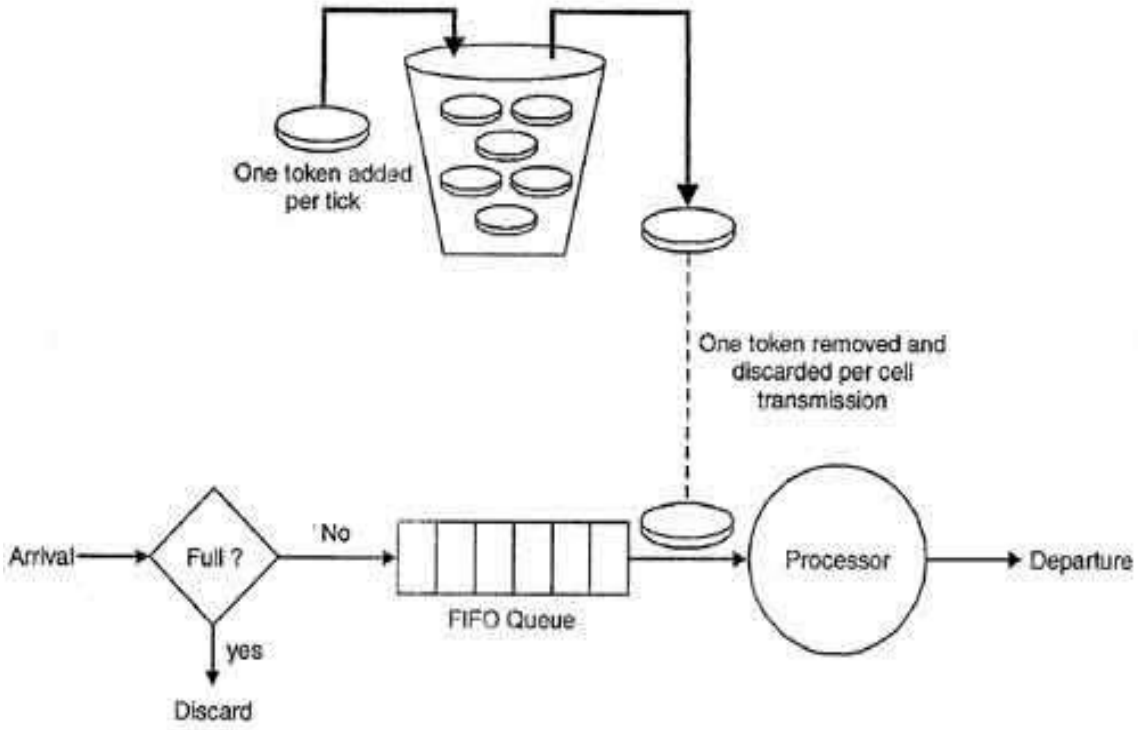
• In this algorithm, a token(s) are generated at every clock tick. For a packet to be transmitted, system must remove token(s) from the bucket.

• Thus, a token bucket algorithm allows idle hosts to accumulate credit for the future in form of tokens.

• For example, if a system generates 100 tokens in one clock tick and the host is idle for 100 ticks. The bucket will contain 10,000 tokens.

Now, if the host wants to send bursty data, it can consume all 10,000 tokens at once for sending 10,000 cells or bytes.

Thus a host can send bursty data as long as bucket is not empty.



Token bucket algorithm