

**VSB ENGINEERING COLLEGE, KARUR – 639 111**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Academic Year: 2017-18(EVEN Semester)**

**CS6601 DISTRIBUTED SYSTEMS**  
**PART A**

**UNIT-I INTRODUCTION**

**1. Define distributed system.**

A distributed system is a collection of independent computers that appears to its users as a single coherent system. A distributed system is one in which components located at networked communicate and coordinate their actions only by passing message.

**2. List the characteristics of distributed system.**

- Programs are executed concurrently
- There is no global time
- Components can fail independently

**3. Mention the challenges in distributed system.**

- Heterogeneity
- Openness
- Security
- Scalability
- Failure handling
- Concurrency
- Transparency

**4. Define heterogeneity.**

The Internet enables users to access services and run applications over a heterogeneous collection of computers and networks. Heterogeneity (that is, variety and difference) applies to all of the following:

- Networks;
- Computer Hardware;
- Operating Systems;
- Programming Languages;
- Implementations By Different Developers.

**5. Why do we need openness?**

The openness of a computer system is the characteristic that determines whether the system can be extended and reimplemented in various ways. The openness of distributed systems is determined primarily by the degree to which new resource-sharing services can be added and be made available for use by a variety of client programs.

**6. Define scalability.**

Distributed systems operate effectively and efficiently at many different scales, ranging from a small intranet to the Internet. A system is described as scalable if it will remain effective when there is a significant increase in the number of resources and the number of users.

**7. What are the types of transparencies?**

Various transparencies types are as follows,

- Access transparency
- Location transparency

- Concurrency transparency
- Replication transparency
- Failure transparency
- Performance transparency
- Scaling transparency

**8. Define transparencies.**

Transparency is defined as the concealment from the user and the application programmer of the separation of components in a distributed system, so that the system is perceived as a whole rather than as a collection of independent components. The implications of transparency are a major influence on the design of the system software.

**9. What are the advantages of Distributed Systems?**

- Performance
- Distribution
- Reliability (fault tolerance)
- Incremental growth
- Sharing of data/resources
- Communication

**10. How we provide a security?**

Confidentiality:

Protection against disclosure to unauthorized individual.

E.g. ACLs (access control lists) to provide authorized access to information.

Integrity:

Protection against alternation or corruption.

E.g. changing the account number or amount value in a money order

Availability:

Protection against interference targeting access to the resources. E.g. denial of service (DoS, DDoS) attacks

Non-repudiation:

Proof of sending / receiving information

E.g. digital signature

**11. What are the different types of system model?**

- Architecture model
- Fundamental model
- Interaction model
- Failure model
- Security model

**12. Define Middleware.**

The term middleware applies to a software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems and programming languages. In addition to solving the problems of heterogeneity, middleware provides a uniform computational model for use by the programmers of servers and distributed applications.

**13. What is the use of middleware?**

Middleware a layer of software whose purpose is to mask heterogeneity and to provide a convenient programming model to application programmers. Middleware is represented by processes or objects in a set of computers that interact with each other to implement communication and resource sharing support for distributed applications.

**14. Define protocol.**

The term protocol is used to refer to a well-known set of rules and formats to be used for communication between processes in order to perform a given task. The definition of a protocol has two important parts to it:

- a. a specification of the sequence of messages that must be exchanged;
- b. a specification of the format of the data in the messages.

**15. What is meant by internet protocol?**

An Internet Protocol (**IP**) **address** is a numerical identification (logical address) that is assigned to devices participating in a computer network utilizing the Internet Protocol for communication between its nodes. Although IP addresses are stored as binary numbers, they are usually displayed in human-readable notations, such as 208.77.188.166 (for IPv4).

**16. Define domain name.**

The **DNS** translates Internet domain and host names to IP addresses. DNS automatically converts the names we type in our Web browser address bar to the IP addresses of Web servers hosting those sites. DNS implements a distributed database to store this name and address information for all public hosts on the Internet. DNS assumes IP addresses do not change (are statically assigned rather than dynamically assigned).

**17. Define mobile IP.**

The Mobile IP protocol allows location-independent routing of IP datagram's on the Internet. Each mobile node is identified by its home address disregarding its current location in the Internet. While away from its home network, a mobile node is associated with a care-of address which identifies its current location and its home address is associated with the local endpoint of a tunnel to its home agent. Mobile IP specifies how a mobile node registers with its home agent and how the home agent routes datagram's to the mobile node through the tunnel.

**18. What is the architectural model?**

Architectural models describe a system in terms of the computational and communication tasks performed by its computational elements; the computational elements being individual computers or aggregates of them supported by appropriate network interconnections. Client- server and peer-to-peer are two of the most commonly used forms of architectural model for distributed systems.

**19. What is the fundamental model?**

Fundamental models take an abstract perspective in order to describe solutions to individual issues faced by most distributed systems. Fundamental model should contain only the essential ingredients that we need to consider in order to understand and reason about some aspects of a system's behavior. The purpose of such a model is:

- To make explicit all the relevant assumptions about the systems we are modeling.
- To make generalizations concerning what is possible or impossible, given those assumptions. The generalizations may take the form of general-purpose algorithms or desirable properties that are guaranteed. The guarantees are dependent on logical analysis and, where appropriate, mathematical proof.

**20. Give examples of distributed systems.**

- Financial trading
- Massively multiplayer online games (MMOGs)
- Web search
- Network of workstations
- Automatic banking (teller machine) system
- Automotive system (a distributed real-time system)
- Distributed Real-Time Systems
- Synchronization of physical clocks

**21. What are the uses of web services?**

A **Web service** is a method of communication between two electronic devices over the Web (Internet). Web services are intrinsically integrated into the World Wide Web, using web standards to represent and discover services.

**22. What are the different types of network?**

The main types of network that are used to support distributed systems: personal area networks, local area networks, wide area networks, metropolitan area networks and the wireless variants of them. Internetworks such as the Internet are constructed from networks of all these types.

**23. Define latency.**

The delay between the start of a message's transmission from one process and the beginning of its receipt by another is referred to as latency.

The latency includes:

- a. The time taken for the first of a string of bits transmitted through a network to reach its destination. For example, the latency for the transmission of a message through a satellite link is the time for a radio signal to travel to the satellite and back.
- b. The time taken by the operating system communication services at both the sending and the receiving processes, which varies according to the current load on the operating systems.

**24. What is meant by networking?**

Networking is the practice of linking two or more computing devices together for the purpose of sharing data. Networks are built with a mix of computer hardware and computer software. Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network.

**25. What is meant by internetworking?**

Internetworking is the practice of connecting a computer network with other networks through the use of gateways that provide a common method of routing information packets between the networks. The resulting system of interconnected networks is called an internetwork, or simply an internet. The most notable example of internetworking is the Internet.

**PART B**

1. Design in detail any two application domain where distributed system is applied.
2. Analyze the challenges in developing a Distributed system and how distributed system overcomes it.
3. Evaluate the trends in distributed system.
4. What are the domain areas in which Distributed System is used?
5. Describe how to compare and contrast cloud computing with more traditional client-server computing? What is novel about cloud computing as a concept?
6. Utilize the World Wide Web as an example to illustrate the concept of resource sharing, client and server.
7. List the three main software components that may fail when a client process invokes a method in a server object, giving an example of a failure in each case. Suggest how the components can be made to tolerate one another's failures
8. (i)Summarize in detail about resource sharing and the challenges involved in it. (ii)Give the types of local resource that are vulnerable to an attack by an untrusted program that is downloaded from a remote site and run in a local computer.

9. Explain in detail about challenges of distributed system.
10. Discuss in detail about the types of transparency
11. Explain about the characteristics of distributed systems.
12. Write about the consequences of distributed systems.
13. Write in detail about the trends of distributed systems.

## UNIT-II COMMUNICATION IN DISTRIBUTED SYSTEM

### PART A

#### 1. What is meant by inter process Communication?

Inter process communication is concerned with the communication between processes in a distributed system, both in its own right and as support for communication between distributed objects. Inter process communication consists of transmitting a message between a socket in one process and a socket in another process. For a process to receive messages, its socket must be bound to a local port and one of the Internet addresses of the computer on which it runs. Messages sent to a particular Internet address and port number can be received only by a process whose socket is associated with that Internet address and port number.

#### 2. What is the difference between RMI and RPC?

In RMI, a calling object can invoke a method in a potentially remote object. As with RPC, the underlying details are generally hidden from the user. RMI allows the programmer to pass parameters not only by value, as input or output parameters, but also by object reference. The remote procedure call approach was a significant breakthrough in distributed systems, providing higher-level support for programmers by extending the concept of a procedure call to operate in a networked environment.

#### 3. Define Datagram.

The term 'datagram' refers to the similarity of this delivery mode to the way in which letters and telegrams are delivered. The essential feature of datagram networks is that the delivery of each packet is a 'one-shot' process; no setup is required, and once the packet is delivered the network retains no information about it. A datagram is, "a self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer.

#### 4. What is the use of UDP?

For some applications, it is acceptable to use a service that is liable to occasional omission failures. For example, the Domain Name System, which looks up DNS names in the Internet, is implemented over UDP. Voice over IP (VOIP) also runs over UDP. UDP datagram's are sometimes an attractive choice because they do not suffer from the overheads associated with guaranteed message delivery.

#### 5. What is meant by client server communication?

The client-server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

#### 6. What is meant by group communication?

Group communication is a multicast operation is more appropriate- this is an operation that sends a single message from one process to each of the members of a group of process, usually in such a way that the membership of the group is transparent to the sender.

#### 7. What is the use of RMI registry?

The RMI registry is used to store a list of available services. A client uses the registry to make its proxy object, and the Registry is responsible for giving appropriate information to the client so that it can hook up with the server that implements the service.

**8. Define Name spaces.**

A name space is the collection of all valid names recognized by a particular service. The service will attempt to look up a valid name, even though that name may prove not to correspond to any object – i.e., to be unbound. Name spaces require a syntactic definition to separate valid names from invalid names. For example, ‘...’ is not acceptable as the DNS name of a computer, whereas www.cdk99.net is valid (even though it is unbound).

**9. Difference between synchronous and asynchronous communication.**

In synchronous form of communication, the sending and receiving processes synchronize at every message. In this case, both send and receive are blocking operations. Whenever a send is issued the sending process is blocked until the corresponding receive is issued. Whenever receive is issued, the process blocks until a message arrives.

In asynchronous form of communication, the use of the send operation is non-blocking in that the sending process is allowed to proceed as soon as the message has been copied to a local buffer and the transmission of the message proceeds in parallel with the sending process. The receive operation can have blocking and non-blocking variants.

**10. What is marshalling and unmarshalling?**

Marshalling is the process of taking a collection of data items and assembling them into a form suitable for transmission in a message. Unmarshalling is the process of disassembling them on arrival to produce an equivalent collection of data items at the destination.

**11. What is XML?**

A “markup language” refers to a textual encoding that represents both a text and details as to its structure or its appearance. XML was designed to describe structured documents and markup languages.

**12. Write about two tier architecture.**

The two tier architecture refers to client/server architecture in which the user interface runs on the client and database is stored on the server. The actual application logic can run on either the client or the server.

**13. What are the common types of failure?**

- Omission failure
- Arbitrary failure
- Timing failure

**14. Define multicast communication.**

Multicast is group communication where information is addressed to a group of destination computers simultaneously.

**15. Define network virtualization.**

Network virtualization refers to the management and monitoring of an entire computer network as a single administrative entity from a single based administrator’s console.

**16. What is the use of middleware?**

Middleware often sits between the operating system and applications on different servers and simplifies the development of applications that leverage services from other applications. This allows programmers to create business applications without having to custom craft integrations for each new application.

Typically, middleware programs provide messaging services so that different applications can communicate using messaging frameworks like Simple Object Access Protocol

(SOAP), Web services, Representational State Transfer (REST) and JavaScript Object Notation (JSON). The systematic tying together of disparate applications, often through the use of middleware, is known as enterprise application integration (EAI).

**17. What is distributed shared memory?**

A distributed shared memory is a mechanism allowing end-users' processes to access shared data without using inter-process communications. In other words, the goal of a DSM system is to make inter-process communications transparent to end-users. Both hardware and software implementations have been proposed in the literature.

- Shared virtual memory
- Object DSM

**18. What are the different types of system models?**

- Physical models
- Architectural models
- Fundamental models
- Interaction models
- Failure models
- Security models

**19. What is the concept of remote procedure call?**

The concept of remote procedure call (RPC) represents a major intellectual breakthrough in distributed computing, with the goal of making the programming of distributed systems look similar, if not identical, to conventional programming – that is, achieving a high level of distribution transparency.

**20. What is the use of distributed shared memory?**

Distributed shared memory (DSM) is an abstraction used for sharing data between computers that do not share physical memory. Processes access DSM by reads and updates to what appears to be ordinary memory within their address space.

**PART B**

1. Explain about remote method invocation. Design how communication is done between Distributed Objects? Explain with the case study Enterprise Java Beans
2. What is RMI? How it is implemented? Write notes on JAVA RMI
3. What is meant by Publish-subscribe systems? Write short notes on it.
4. Describe With a neat sketch explain remote procedure call.
5. Describe explain about group communication.
6. Discuss in detail the physical & architectural model with necessary diagrams.
7. Describe on the fundamental models with example and diagrams.
8. What is the purpose of external data representation and marshaling?
9. Examine about Multicast communication.
10. Explain in detail the following (i) Middleware layers (ii) UDP datagram communication (iii) TCP stream communication. (iv) Characteristics of Inter process communication.
11. Define Overlay networks. What are the types of overlay networks? Explain in detail.
12. Examine about the use of request reply protocol.
13. Show how message queues are useful? Explain in brief.
14. Illustrate what is shared memory? Explain its approaches with example
15. Analyze in detail about Message Passing Interface (MPI).

## UNIT-III PEER TO PEER SERVICES AND FILE SYSTEM

### PART A

**1. What is routing overlay?**

A routing overlay is a distributed algorithm for a middleware layer responsible for routing requests from any client to a host that holds the object to which the request is addressed.

**2. Define peer-peer communication.**

Peer to peer is a decentralized communication model in which each party has the same capabilities and either party can initiate a communication session.

**3. Explain characteristics of peer-to-peer systems.**

- Global scalability
- Load balancing
- Local optimization
- Security of data
- Adjusting to dynamic host availability

**4. Give the features of peer to peer systems.**

- Large scale sharing of data and resources
- No need for centralized management
- All the nodes in a peer to peer system have the same functional capabilities and responsibilities.

**5. What is Napster?**

Napster was developed for peer to peer file sharing especially MP3 files. They are not fully peer to peer since it used central servers to maintain lists of connected systems and the files they provided, while actual transactions were conducted directly between machines.

**6. Define pastry.**

Pastry is a generic, scalable and efficient for peer to peer applications. Pastry nodes form a decentralized, self organizing and fault tolerance overlay network within the internet.

**7. Give the capabilities of pastry.**

- mapping application objects to pastry nodes
- Inserting objects
- Accessing objects
- Availability and persistence
- Diversity
- Load balancing

**8. What is DFS?**

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN).

**9. What is LDAP?**

LDAP is a standard technology for building computer directory is a specialized database that stores information about devices, applications, people and other aspects of a computer network.

**10. What is absolute name and relative name?**

An absolute name begins at the root context of the name space tree and follows a path down to a specified object, giving the context names on the path.

A relative name defines a path from the current context to the specified object.

**11. What is stateful server and stateless server?**

A stateful server maintains information about all clients that are utilizing the server to access a file.

A stateless server maintains no client information. Each and every request from a client must include very specific request information such as file name, operation.

**12. What are the components in file structure?**

File service architecture offers a clear separation of the main concerns in providing access to files is obtained by structuring the file service as three components:

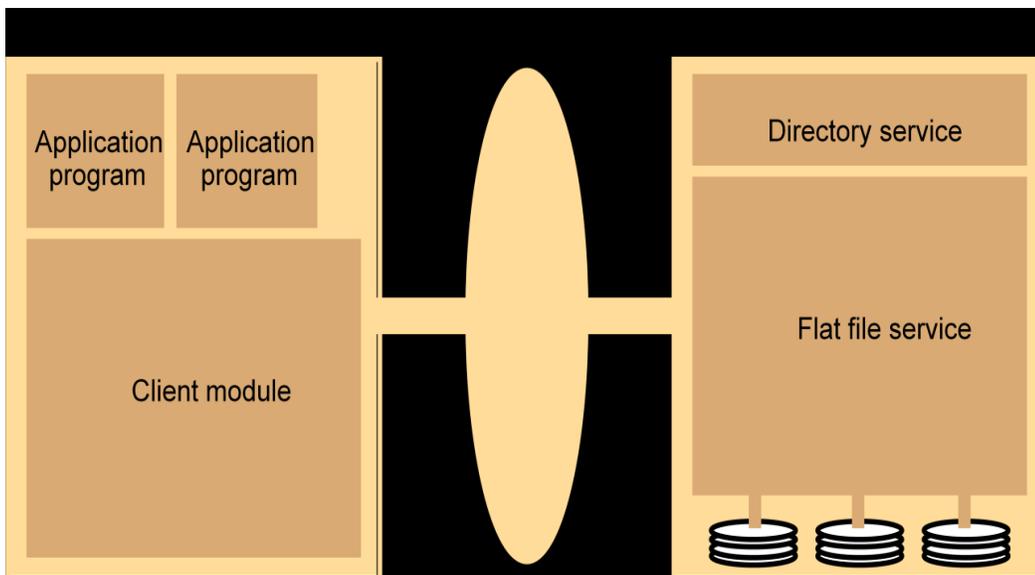
- A flat file service
- a directory service
- A client module

**13. What are the layers in name space?**

Three layers used to implement such distributed name spaces

- Global layer
- Administrational layer
- Managerial layer

**14 .Sketch the file service architecture?**



**15. Define name space.**

A namespace is the collection of all valid names recognized by a particular service. The service will attempt to look up a valid name, even though that name may prove not to correspond to any object. Name space requires a syntactic definition to separate.

**16. What is meant by zone?**

The DNS naming data are divided into zones. A zone contains the following data:

- Attributes data for names in a domain, less any subdomains administrates by lower level authorities.
- The names and address of at least two name servers that provide authoritative data for the zone. These are versions of zone data that can be relied upon as being reasonably up to date.

- The names of name servers that hold authoritative data for delegated sub domains; and 'glue' data giving the IP address of these servers.
- Zone-management parameters, such as those governing the catching and replication of zone data.

### **17. What is meant by navigation?**

The process of locating naming data from than more than one name server in order to resolve a name is called navigation. The client name resolution software carries out navigation on behalf of the client. It communicates with name servers as necessary to resolve a name.

### **18. What is multicast navigation?**

In multicast navigation, a client multicast the name to be resolved and required objects type to the group of name servers. Only the server that holds the named attributes responds to the request. Unfortunately, however, if the name proves to be unbound, the request is greeted with silence.

### **19. What is meant by directory services?**

The directory services provide a mapping between text names for files and their UFIDs. Client may obtain the UFIDs of a file by quoting its text name to the directory services. The directory services provide the function needed to generate directories, to add new file name to directories and to obtain UFIDs from directories. It is client of the flat file services; its directory is stored in files of the flat services. When a hierarchic file-naming scheme is adopted as in UNIX, directories hold references to other directories.

### **20. What is file replication?**

In a file service that supports replication, a file may be represented by several copies of its contents at different locations. This has two benefits-its enables multiple servers to share the load of providing a service to clients accessing the same set of files, enhancing the scalability of the service, and it enhances fault tolerance by enabling clients to locate another server that holds a copy of the file when one has failed. Few file services support replication fully, but most support the catching of files or portions of files locally, a limited form of replication.

## **PART B**

1. Define Peer to Peer systems. Explain in detail the working of Peer to Peer Systems.
2. What is meant by Napster legacy? Explain.
3. Explain about File system access model and its sharing semantics.
4. Explain on Peer to Peer middleware systems.
5. Illustrate with a case study explain about the application of distributed algorithm Routing overlays.(Pastry/Tapestry)
6. Analyze in detail about Distributed File system, its characteristics and requirements.
7. Describe the working of File service architecture.
8. Summarize the purpose of Andrew File system
9. Formulate the design and implementation of name services and Domain Name services.
10. Discuss the architecture and server operation of NFS.
11. Illustrate in brief about LDAP.
12. List the different approaches to implement the Name Caches and explain them briefly.

**UNIT-IV SYNCHRONIZATION AND REPLICATION**  
**PART A**

**1. Define clock skew, clock drift and clock drift rate.**

Clock skew is defined as the difference between the times in two clocks.

Clock drift is the count time at different rates.

Clock drift rate is the difference in precision between a perfect reference clock and a physical clock.

**2. Give the types of clocks.**

Two types of clocks are used:

- Logical clocks: to provide consistent event ordering.
- Physical clocks: clocks whose values must not deviate from the real time by more than a certain amount.

**3. What are the techniques used to synchronize clocks?**

- Time stamps of real time clocks
- message passing
- round trip time

**4. Define causal ordering.**

The partial ordering obtained by generalizing the relationship between two processes is called causal ordering or potential causal ordering.

**5. What is global state?**

The global state of a distributed system consists of the local state of each process together with the messages that are currently in transit that is that have been sent but not delivered.

**6. Define distributed deadlock.**

A distributed deadlock occurs when each of a collection of processes waits for another process to send it a message and where there is a cycle in the graph of this wait-for relationship.

**7. What is mutual synchronization?**

This exploits mutual exclusion between peer processes based upon multicast. Processes that require entry to a critical section multicast a request message and can enter it only when all the other processes have replied to this message.

**8. What is election algorithm?**

An algorithm for choosing a unique process to play a particular role is called an election algorithm.

**9. Define transaction.**

A transaction defines a sequence of server operations that is guaranteed to be atomic in the presence of multiple clients and server crash.

**10. List the methods to ensure serializability.**

There are three ways to ensure serializability.

- Locking
- Timestamp ordering
- Optimistic concurrency control.

**11. What is distributed debugging?**

Distributed debugging is nothing but to check whether a transitory state, instead of a stable state has occurred in an actual execution. This is done by recording a system global state.

**12. What is logical clock?**

Logical clock is a monotonically increasing software counter, whose value need bear no particular relationship to any physical clock. Each process keeps its own logical clock which it uses to apply so called lamport timestamps to events.

**13. What is vector clock?**

Vector clock are used in a distributed system to determine whether pairs of events are causally related. Using vector clocks, timestamps are generated for each event in the system and their causal relationship is determined by comparing those timestamps.

**14. What is mutual exclusion?**

Mutual exclusion in a distributed system states that only one process is allowed to execute the critical section at any given time. In a distributed system, shared variables or a local kernel cannot be used to implement mutual exclusion.

**15. What are the requirements of mutual exclusion algorithms?**

- Freedom from deadlocks
- Freedom from starvation
- Strict fairness
- Fault tolerance

**16. What is network partition?**

The network partition can be used to separate a group of replica managers into two or more subgroups. The members of same subgroup communicate with one another but members of different subgroup cannot communicate with one another.

**17. Define multicast communication.**

It is the implementation of group communication. Multicast communication requires coordination and agreement. The aim is for members of a group to receive copies of messages sent to the group.

**18. What is nested transaction? Give an example.**

A nested transaction is used to provide a transactional guarantee for a subset of operations performed within the scope of a larger transaction. CORBA object transaction service supports both flat and nested transactions.

**19. What is the use of locks?**

Serializing mechanism uses exclusive locks. In this locking scheme, the server attempts to lock any object that is about to be used by any operation of a client's transaction.

**20. What are the types of locks?**

- Read locks
- Write locks
- Commit locks

**PART B**

1. Describe in detail about Cristian's and Berkeley algorithm for synchronizing clocks.
2. Design Flat transaction and nested transaction with example.
3. Explain detail about two phase commit protocol.
4. What a short note on Ricart and Agrawala's algorithm & Maekawa's voting algorithm.
5. Examine how mutual exclusion is handled in distributed system.
6. Discuss that Byzantine agreement can be reached for three generals, with one of them faulty, if the generals digitally sign their messages
7. Examine a solution to reliable, totally ordered multicast in a synchronous system, using a reliable multicast and a solution to the consensus problem
8. Describe about distributed dead locks.
9. Examine briefly about optimistic concurrency control.

## UNIT-V PROCESS & RESOURCE MANAGEMENT

### PART A

**1. What is process migration?**

The phenomenon of shifting a process from one machine to another one which is called process migration. It is the act of transferring process between two machines.

**2. What is task assignment approach?**

In a task assignment approach, each process submitted by a user for processing is viewed as a collection of related tasks and these tasks are scheduled to suitable nodes so as to improve performance.

**3. What is DSM?**

Distributed shared memory (DSM) is an abstraction used for sharing data between computers that do not share physical memory. Processes access DSM by reads and updates to what appears to be ordinary memory within their address space. DSM is primarily a tool for parallel applications or for any distributed application or group of applications in which individual shared data items can be accessed directly.

**4. What is weak consistency?**

Synchronization accesses (accesses required to perform synchronization operations) are sequentially consistent. Before synchronization access can be performed, all previous regular data accesses must be completed. Before a regular data access can be performed, all previous synchronization accesses must be completed. This essentially leaves the problem of consistency up to the programmer. The memory will only be consistent immediately after a synchronization operation.

**5. What is release consistency?**

Release consistency is essentially the same as weak consistency, but synchronization accesses must only be processor consistent with respect to each other. Synchronization operations are broken down into acquire and release operations. All pending acquires (e.g., a lock operation) must be done before a release (e.g., an unlock operation) is done. Local dependencies within the same processor must still be respected.

"Release consistency is a further relaxation of weak consistency without a significant loss of coherence."

**6. What is Casual Consistency Model?**

"The causal consistency model relaxes the requirement of the sequential model for better concurrency. Unlike the sequential consistency model, in the causal consistency model, all processes see only those memory reference operations in the same (correct) order that are potentially causally related. Memory reference operations that are not potentially causally related may be seen by different processes in different orders."

**7. What is Strict Consistency Model?**

"The strict consistency model is the strongest form of memory coherence, having the most stringent consistency requirements. A shared-memory system is said to support the strict consistency model if the value returned by a read operation on a memory address is always the same as the value written by the most recent write operation to that address, irrespective of the locations of the processes performing the read and write operations. That is, all writes instantaneously become visible to all processes."

**8. What is thread pool?**

Thread pool consists of a number of threads in a pool where they await work. This is slightly faster to service a request with an existing thread than create a new thread. This allows the number of threads in the application to be bound to the size of the pool.

**9. What is multithreaded program?**

A multithreaded program contains two or more parts that can run concurrently and each part can handle different task at the same time making optimal use of the available

resources especially when your computer has multiple CPUs.

**10. What is threshold policy?**

This policy selects a random node, checks whether the node is able to receive the process and then transfers the process. If node rejects, another node is selected randomly. This continues until probe limit is reached.

**11. What is single threshold and double threshold policy?**

Single threshold policy may lead to unstable algorithm because under loaded node could turn to be overloaded right after a process migration.

When node is in overloaded region new local processes are sent to run remotely, requests to accept remote processes are rejected. When node is in normal region new local processes run locally, requests to accept remote process are requested.

**12. What is load sharing approach?**

It is necessary and sufficient to prevent nodes from being idle while some other nodes have more than two processes. Load sharing is much simpler than load balancing since it only attempts to ensure that no node is idle when heavily loaded node exists.

**13. What are the two update options?**

Write-update: The updates made by a process are made locally and multicast to all other replica managers possessing a copy of the data item, which immediately modify the data read by local processes. Processes read the local copies of data items, without the need for communication. This is also known as multiple-reader/multiple-writer sharing.

Write-invalidate: This is commonly implemented in the form of multiple-reader/single-writer sharing. When a process attempts to write to it, a multicast message is first sent to all other copies to invalidate them and this is acknowledged before the write can take place; the other processes are thereby prevented from reading stale data (that is, data that are not up to date). Any processes attempting to access the data item are blocked if a writer exists.

**14. What is thrashing?**

Thrashing is said to occur where the DSM runtime spends an inordinate amount of time invalidating and transferring shared data compared with the time spent by application processes doing useful work. It occurs when several processes compete for the same data item, or for falsely shared data items.

**15. What is CORBA?**

CORBA is a middleware design that allows application programs to communicate with one another irrespective of their programming languages, their hardware and software platforms, the networks they communicate over and their implementers.

**16. What is CORBA naming service?**

It is a binder that provides operations including rebind for servers to register the remote object references of CORBA objects by name and resolve for clients to look them up by name. The names are structured in a hierarchic fashion, and each name in a path is inside a structure called a Name Component.

**17. What is load balancing?**

Load balancing is nothing but the allocation of tasks or jobs to processors to increase overall processor utilization and throughput. Actually load balancing is done by process migration.

**18. List desirable features of good process migration mechanism.**

Features are transparency, efficiency, minimal interference, minimize freezing time and minimal residual dependencies.

**19. Define the term thread.**

A minimal software processor in whose context a series of instructions can be executed. Saving a thread context implies stopping the current execution and saving all the data needed to continue the execution at a later stage.

**20. Explain the benefit of process migration.**

- Better response time and execution speed up.
- Reducing network traffic
- Improving system reliability
- Higher throughput and effective resource utilization.

**PART B**

1. Describe in detail the features involved in process migration.
2. Evaluate on the mechanism used in process migration.
3. Explain how process migration is implemented in heterogeneous system?
4. Illustrate in detail about threads and its process.
5. Discuss the purpose of task assignment approach.
6. Describe about the Load Balancing approach in detail.
7. Summarize in detail how the load is being shared among the resources.
8. Desirable features of scheduling algorithm?
9. Write about process migration.
10. Discuss in detail about load balancing algorithms.
11. Write about load balancing policies.

# IT6601 Mobile Computing

## PART A UNIT I INTRODUCTION

1. What is mobile computing?

Mobile computing is a technology that allows transmission of data, via a computer, without having to be connected to a fixed physical link.

2. Difference between FDMA and TDMA.

FDMA	TDMA
Frequency Division Multiplexing Access is a method of dividing a single band into 30 discrete channels.	Time Division Multiplexing Access that also allows more subscribers to use the same frequency band.
Each channel would then be capable of handling separate traffic, whether it's a call or a data transfer.	TDMA divides a single channel into 3 discrete time partitions.
FDMA is also used in the Total Access Communication System (TACS).	TDMA is utilized by Digital-Advanced Mobile Phone System (D-AMPS) and Global System for Mobile communications (GSM).

3. Why do Hidden and Exposed terminal problem arise.

Hidden terminal problem is due to the fact that a node (say A) transmitting to another node (say B) cannot hear transmissions from another node C, which might also be transmitting to B, and might interfere with the A-to-B transmissions.

Exposed node problem occurs when a node is prevented from sending packets to other nodes because of a neighboring transmitter.

4. Prove that Barker code has good auto correlation.

When the receiver attempts to correlate the received coded symbols with respect to any of the codes which it internally generates, it is not able to correlate even when it uses exactly the same code as the one used for transmission.

Reasons for no correlation:

- Propagation delay
- Inappropriate code

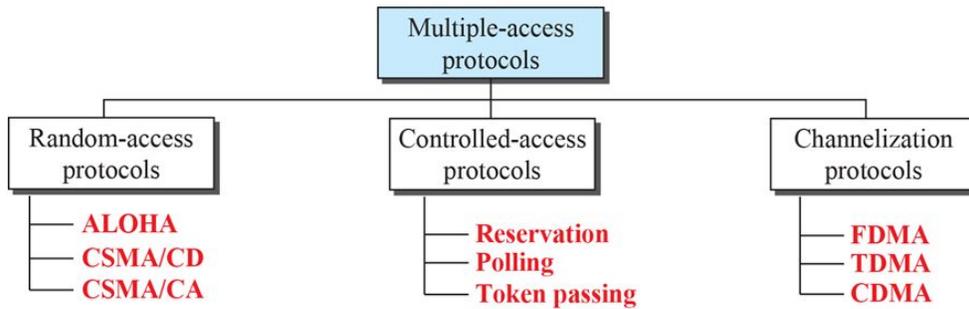
5. Write the characteristics of mobile computing.

- Ubiquity
- Location awareness
- Adaptation
- Broadcast
- Personalization

6. Draw the 3 Tier structure of Mobile computing application.

Presentation(Tire-1)
Application(Tier-2)

7. State the issue in wireless MAC protocol.
  - Hidden Terminal Problem.
  - Exposed Terminal Problem.
8. Draw the Taxonomy of MAC protocol.



9. Write the different Random assignment scheme in MAC.
  - ALOHA
  - Slotted ALOHA
  - CSMA
  - CSMA/CD
  - CSMA/CA

10. Difference between Mobile Computing vs Wireless Networking.

<b>Mobile Computing</b>	<b>Wireless Networking</b>
Mobile Computing denotes accessing information and remote computational services.	Wireless Networking provides the basic communication infrastructure.
It is based on wireless networking.	Wireless Networking is an important ingredient of mobile computing.

11. Define mobile IP.

Mobile IP (or MIP) is an Internet Engineering Task Force (IETF) standard communications protocol that is designed to allow mobile device users to move from one network to another while maintaining a permanent IP address.

12. What is Tunneling process?

The encapsulation process creates a logical construct called a tunnel between the device that encapsulates and the one that decapsulates.

13. Define COA or how the mobile nodes identify the current location? Explain.

The mobile nodes identify the current location with the COA. A care-of address (usually referred to as CoA) is a temporary IP address for a mobile device. This allows a home agent to forward messages to the mobile device. A separate address is required because the IP address of the device that is used as host identification is topologically incorrect - it does not match the network of attachment. The care-of address splits the dual nature of an IP address, that is, its use is to identify the host and the location within the global IP network.

14. State the features of mobile IP.
- Transparency
  - Compatibility
  - Security
  - Efficiency and Scalability
15. State the key mechanism used in Mobile IP.
- Discovering the care-of-address
  - Registering the care-of-address
  - Tunnelling to the care-of-address
16. What are applications of Mobile Computing?
- Vehicles
  - Emergencies
  - Business
  - Replacement of wired networks
  - Infotainment
  - Location dependent services
  - Mobile and wireless devices
17. What are the obstacles in mobile communications?
- Interference
  - Regulations and spectrum
  - Low Bandwidth
  - High delays, large delay variation
  - Lower security, simpler to attack
  - Shared Medium
  - Adhoc-networks
18. What are the Advantages of wireless LAN?
- Flexibility
  - Planning
  - Design
  - Robustness

19. Mention some of the disadvantages of WLANS?
- Quality of service
  - Proprietary solutions.
  - Restrictions
  - Safety and Security

20. Ubiquitous computing

Ubiquitous computing enhances computer use by making many computers available throughout the physical environment, while making them effectively invisible to users.

21. What is the aim of ubiquitous computing?

The aim of ubiquitous computing is to design computing infrastructures in such a manner that they integrate seamlessly with the environment and become almost invisible.

It integrates computation into the environment, rather than having computers which are distinct objects

## **PART B**

1. Explain in detail about the various generations of wireless communication system.
2. Explain in detail about the characteristics of Mobile computing.
3. Explain about the structure of mobile computing applications.
4. Explain Wireless MAC issues with suitable diagram / Summarize the issues of Wireless MAC Protocols.
5. Explain in detail about the Fixed assignment schemes.
6. Explain in detail about the Random assignment schemes.
7. Explain in detail about Mobile Computing. Classify its various applications in the real world scenario.

### **UNIT II MOBILE INTERNET PROTOCOL AND TRANSPORT LAYER**

1. What are the requirements of mobile IP?

- Compatibility
- Transparency
- Scalability and efficiency
- Security

2. Mention the different entities in a mobile IP.

- Mobile Node
- Correspondent Node
- Home Network
- Foreign Network
- Foreign Agent
- Home Agent
- Care-Of address
- Foreign agent COA
- Co-located COA

3. Define Mobile node:

A mobile node is an end-system or router that can change its point of attachment to the Internet using mobile IP. The MN keeps its IP address and can continuously with any other system in the Internet as long as link layer connectivity is given.

4. Explain Cellular IP.

Cellular IP provides local handovers without renewed registration by installing a single cellular IP gateway for each domain, which acts to the outside world as a foreign agent.

5. What do you mean by mobility binding?

The Mobile Node sends its registration request to the Home Agent. The HA now sets up a mobility binding containing the mobile node's home IP address and the current COA.

6. Define COA.

The COA (care of address) defines the current location of the MN from an IP point of view. All IP packets sent to the MN are delivered to the COA, not directly to the IP address of

the MN. Packet delivery toward the MN is done using the tunnel. DHCP is a good candidate for supporting the acquisition of Care Of Addresses.

7. Define a tunnel.

A tunnel establishes a virtual pipe for data packets between a tunnel entry and a tunnel endpoint. Packets entering a tunnel are forwarded inside the tunnel and leave the tunnel unchanged.

8. What is encapsulation?

Encapsulation is the mechanism of taking a packet consisting of packet header and data putting it into the data part of a new packet.

9. What is decapsulation?

The reverse operation, taking a packet out of the data part of another packet, is called decapsulation.

10. Define Snooping TCP.

A protocol in which an agent buffers the packets from the fixed connection layer for transmission to the mobile node on a wireless transceiver; the agent also buffers the packets on the wireless transceiver from the node for transmitting to a layer at the fixed line. The agent snoops at the transmission and reception in place of acknowledgement-or-timeout-based TCP method in the mobile part of the network.

11. Define Mobile TCP.

A method of splitting the TCP layer into two TCP sub-layers using a mechanism that reduces window size to zero. The split is asymmetric; The window is set to zero to prevent the transmission from the TCP transport layer at the mobile node (MN) or at the fixed node when disconnection is noticed. The window opens again on getting the packet, there is no slow start by the base transceiver and it is presumed that packet loss is due to disconnection and not due to congestion or interference.

12. Write the concept of “Fast Retransmit/ Fast Recovery Transmission”.

A method in which there are four or more phases of fast retransmit and fast recovery – first phase as slow start and beginning (exponential), then fast retransmit/recovery phase 1 (FRR1) on three duplicate acknowledgements, fast retransmit/fast recovery phase 2 (FRR2), and wait (Constant time out and window size).

13. Define T-TCP.

A protocol which is efficient and is used in situations where short messages are to be sent in sequence and a packet is delivered after the SYN and SYN\_ACK packet exchanges and the connection closes after the packet exchanges of FIN, FIN\_ACK, and CLOSING.

14. List the features of TCP.

The main features of TCP are:

- 1) Transmission as data Streams
- 2) Buffering and retransmission
- 3) Session-start, data transfer, and session-finish fully acknowledged end to end.
- 4) In-order delivery
- 5) Congestion Control and avoidance

15. What is explicit notification?

A method of congestion control by explicit notification of congestion, for example, when a base transceiver at the receiver end is not able to transmit a packet to the mobile node then it

sends an ESNB (explicit bad state notification) to the sender (on fixed line ) at the other end.

16. What is selective retransmission?

A method in which there is an additional acknowledgement, known as selective acknowledgement; a timeout is set at transmitting end for receiving SACKs. Only the lost packet corresponding to a SACK needs to be retransmitted.

17. What is TCP header?

A header used in the TCP protocol; it consists of fields in five 32-bit words followed by words for the option fields and padding.

18. Methods of Congestion Control.

The methods of congestion control:

- 1) Slow start and congestion avoidance
- 2) Fast recovery after packet loss
- 3) Fast retransmit and fast recovery
- 4) Selective acknowledgement
- 5) Explicit congestion notification

19. What are Advantage and Disadvantage of Mobile TCP?

**Advantages:**

- i. M-TCP maintains the TCP end-to-end semantic. The SH does not send any ACK itself but forwards the ACKs from the MH.
- ii. If the MH is disconnected, M\_TCP avoids useless retransmissions, slow starts or breaking connections by simply shrinking the sender's window to 0;
- iii. Since M-TCP does not buffer data in the SH as I-TCP does, it is not necessary to forward buffers to a new SH. Lost packets will be automatically retransmitted to the new SH.

**Disadvantages:**

- i. As the SH does not act as proxy as in I-TCP, packet loss on the wireless link due to bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption.
- ii. A modified TCP on the wireless link not only requires modification to the MH protocol software but also new network elements like the bandwidth manager

20. What led to the development of Indirect TCP?

TCP performs poorly together with wireless links

TCP within the fixed network cannot be changed. This led to the development of I-TCP which segments a TCP connection into a fixed part and a wireless part.

21. What is the goal of M-TCP?

The goal of M-TCP is to prevent the sender window from shrinking if bit errors or disconnection but not congestion cause current problems. It wants

- To provide overall throughput
- To lower the delay
- To maintain end-to-end semantics of TCP
- To provide a more efficient handover.

22. What do you mean by persistent mode?

Persistent mode is the state of the sender will not change no matter how long the receiver is disconnected. This means that the sender will not try to retransmit the data.

### **PART B**

1. What is Mobile IP? Explain about the terminologies of Mobile IP.
2. Explain about the features and key mechanisms of Mobile IP.
3. Express brief account of route optimization in Mobile IP.
4. Explain in detail about the Improvement in TCP performance.

## **UNIT-III MOBILE TELECOMMUNICATION SYSTEM**

### **PART A**

1. Define GSM?

The global system for mobile communication (GSM) was developed by Groupe Speciale Mobile (GSM) which was founded in Europe in 1992. The GSM is a standard for mobile telecommunication through a cellular network at data rates if up to 14.4 kbps. Nowadays, it consists of a set of standards and protocols for mobile telecommunication.

2. Define GPRS?

General Packet Radio Service (GPRS) is a packet oriented service for mobile devices data communication which utilizes the unused channels in TDMA mode in a GSM network and also sends and receives packet of data through the internet.

3. What are subsystems in GSM system?

Radio subsystem (RSS)

Network & Switching subsystem (NSS) Operation subsystem (OSS)

4. What are the control channel groups in GSM?

The control channel groups in GSM are:

Broadcast control channel (BCCH)

Common control channel (CCCH)

Dedicated control channel (DCCH)

5. What are the four types of handover available in GSM?

- Intra cell Handover
- Inter cell Intra BSC Handover
- Inter BSC Intra MSC handover
- Inter MSC Handover

6. What is the frequency range of uplink and downlink in GSM network?

The frequency range of uplink in GSM network is 890-960 MHz

The frequency range of downlink in GSM network is 935-960 MHz

7. What are the security services offered by GSM?

The security services offered by GSM are:

- Access control and authentication.
- Confidentiality.
- Anonymity.

8. What are the reasons for delays in GSM for packet data traffic?

Collisions only are possible in GSM with a connection establishment. A slotted ALOHA mechanism is used to get access to the control channel by which the base station is told about the connection establishment attempt. After connection establishment, a designated channel is installed for the transmission.

9. What is meant by beacon?

A beacon contains a timestamp and other management information used for power management and roaming. e.g., identification of the base station subsystem (BSS)

10 . List out the numbers needed to locate an MS and to address the MS.

The numbers needed to locate an MS and to address the MS are: Mobile station international ISDN number (MSISDN) International mobile subscriber identity (IMSI)

Temporary mobile subscriber identity (TMSI) Mobile station roaming number (MSRN)

11 . What is meant by GPRS?

The General Packet Radio Service provides packet mode transfer for applications that exhibit traffic patterns such as frequent transmission of small volumes.

12. What is meant by GGSN?

GGSN is Gateway GPRS Support Node. It is the inter-working unit between the GPRS network and external packet data networks. The GGSN is connected to external networks via the GI interface and transfers packets to the SGSN via an IP based GPRS backbone network.

13. What is meant by SGSN?

SGSN is Serving GPRS Support Node. It supports the MS via the GB interface. The GSN is connected to a BSC via frame relay.

14. What is meant by BSSGP?

BSSGP is Base Station Subsystem GPRS Protocol. It is used to convey routing and QoS-related information between the BSS and SGSN. BSSGP does not perform error correction and works on top of a frame relay network.

15. Expand GSM, GPRS and UMTS.

- Global System for Mobile Communication (GSM)
- General Packet Radio Service (GPRS)
- Universal Mobile Telecommunication System (UMTS)

16. Write the GSM services.

- Supplementary Services
- Bearer Services
- Tele Services

17. What is RSS?

- RSS stands for Radio subsystem (RSS)
- RSS comprises all radio specific entities

18. Name the entities of RSS.

- Base Station Subsystem (BSS)
- Base Transceiver Station (BTS)
- Base Station Controller (BSC)
- Mobile Station (MS)

19. Mention the advantages of GSM.
- o Communication
  - Total mobility
  - Worldwide connectivity
  - High capacity
  - High transmission quality'
  - Security functions
20. What does SIM card contain?
- a personal identity number (PIN)
  - a PIN unblocking key (PUK)
  - an authentication key Ki
  - the international mobile subscriber identity (IMSI)
21. Mention the disadvantages of GSM.
- No end-to-end encryption of user data
  - Reduced concentration while moving
  - Electromagnetic radiation
  - High complexity of system
  - Several incompatibilities within the GSM standards
  - Card-type
  - Serial number
  - A list of subscribed services
22. Write the congestion avoidance algorithm.
- TCP Tahoe and Reno
  - TCP Vegas
  - TCP New Reno
23. State the mechanism used in improvement of TCP Performance.
- Slow start
  - Mobile TCP
  - Indirect TCP
  - Snooping TCP

### **PART B**

1. Explain the architecture of GSM in detail.
2. Explain the architecture of GPRS in detail.
3. Explain in detail about the UMTS Network architecture
4. Summarize 1G,2G,3G cellular networks in detail.
5. Demonstrate briefly about VHE. In what way is VHE is applied in 3G networks?
6. Show how a GSM network Provides security to the customers

## **UNIT-IV MOBILE AD-HOC NETWORKS**

### **PART A**

1. What is Ad-hoc network?

The meaning for Ad-hoc is “For the purpose”. The network that formed for the particular purpose without infrastructure is called as Ad-hoc network.

2. List the categories of Ad-hoc networks.
  - Mobile Ad-hoc Network
  - Vehicular Ad-hoc Network
  - Sensor Network
  
3. What are the characteristics of MANET?
  - Lack of fixed Infrastructure
  - Dynamic topologies
  - Bandwidth constrained, variable capacity link
  - Energy constrained operations
  - Increased vulnerability
  
4. What are the applications of MANET?
  - Communication among portable computers
  - Environmental monitoring
  - Military
  - Emergency applications
  
5. What are the design issues of MANET?
  - Network size and node density
  - Connectivity
  - Network topology
  - User traffic
  - Operational environment
  - Energy constraint
  
6. State the essential routing protocols.
  - Link state protocols
  - Distance vector protocols
  
7. List the types of communication in MANET.
  - Unicast
  - Multicast
  - Broadcast
  
8. Classification of unicast routing protocols.
  - Proactive protocols
  - Reactive Protocols
  - Hybrid protocols
  
9. List the popular routing protocols in MANET.
  - Destination sequence distance vector routing protocol
  - Dynamic source routing protocol
  - Ad-hoc on-demand distance vector
  - Zone routing protocol
  
10. Mention the classification of multicast routing protocols.

- Tree based protocol
- Mesh based protocol

11. Define VANET.

A vehicular Ad-hoc Network is a special type of MANET in which moving automobiles form the nodes of the network.

12. What are the security issues in MANET?

- Lack of physical boundary
- Low power RF transmissions
- Limited computational capabilities
- Limited power supply

13. What are the characteristics of secure ad-hoc networks?

- Availability
- Confidentiality
- Integrity
- Authentication
- Non-repudiation

14. What are the types of attack in MANET?

- Active attack
- Passive attack

15. Define Active attack. Give example.

The attack is destructive and disturbs the normal functionality of the network.

Example: wormhole, black hole

16. Define Passive attack. Give example.

The attack that monitors and steals the data exchanged the network, without disturbing the network operation.

Example: Snooping, Eavesdropping

17. Differentiate cellular and Ad-Hoc Networks:

Cellular Network	Ad-Hoc Network
Infrastructure Networks	Infrastructure less Networks.
Fixed, pre-located cell sites and base stations	No base station, and rapid deployment
Static backbone network topology	Highly dynamic network topologies
Relatively caring environment and stable connectivity	Hostile environment and irregular connectivity
Detailed planning before base station can be installed	Ad-Hoc network automatically forms and adapts to changes
High setup costs	Cost-effective
Large setup time	Less setup time

## 18. Compare MANET Vs VANET

MANET	VANET
MANET - Mobile Adhoc NETWORK	VANET- Vehicular Adhoc NETWORK
Nodes moves randomly	Nodes moves regularly
Mobility is low	Mobility is high
Reliability is medium	Reliability is high
Node lifetime depends on power source	Node lifetime depends on vehicle life time
Network topology is sluggish and slow	Network topology is frequent and fast

## 19. List any five attacks in MANET.

- Snooping,
- Eavesdropping
- wormhole,
- Black hole
- Monitoring

## 20. Write the type of network used in moving automobile.

The type of network used in moving automobile is VANET.

VANET- The Vehicular Ad-Hoc Network.

VANET, is a technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range.

## 21. List the Types of Communications.

- Unicast  
Message is sent to a single destination node
- Multicast  
Message is sent to a selected subset of network nodes
- Broadcast  
Broadcasting is a special case of multicasting  
Message is sent to all the nodes in the network

## 22. What are the functions of each node in MANET?

Forward the packet to the next hop

Before forwarding, Sender has to ensure that:

- The packet moves towards its destination
- The number of hops(path length) to destination is minimum
- Delay is minimized
- Packet loss is minimum through the path
- Path does not have a loop

## 23. Discuss the Proactive (table-driven) protocols.

- It is also known as table-driven routing protocols
- Each node in the routing table maintains information about routes to every other node in

- the network.
  - Tables are updated frequently due to
    - Changes in network topology
    - Node Movements
  - Nodes shutting down
    - Nodes can determine the best route to a destination
  - Generates a large number of control messages to keep the routing tables up-to-date
    - Generates overhead which consumes large part of available bandwidth
24. Define Reactive protocols.
- It is also called as On-demand routing protocol
  - Nodes do not maintain up-to-date routing information
    - New routes are discovered only when required
  - Uses flooding technique to determine the route
    - Flooding technique is used when the node does not have routing knowledge

### **PART B**

1. Explain in detail about the characteristics of MANET.
2. Explain in detail about the design issues of MANET.
3. Explain about the various applications of MANET.
4. Explain in detail about the classification of unicast MANET Routing protocols.
5. Define short notes on: a) Characteristics of a secure ad hoc network. b) Security attacks in ad hoc networks.
6. Explain about the various attacks on Ad- hoc networks.

## **UNIT-V MOBILE PLATFORMS AND APPLICATIONS**

### **PART A**

1. What are the Responsibilities of OS in Mobile Devices?
  - Managing Resources
  - Providing different interfaces
2. What are the special constraints of Mobile OS?
  - Limited memory
  - Limited screen size
  - Miniature keyboard
  - Limited processing power
  - Limited battery power
  - Limited and fluctuating bandwidth of the wireless medium.
3. What are the special service requirements of Mobile OS?
  - Support for specific communication protocols
  - Support for a variety of input mechanisms
  - Compliance with open standards
  - Extensive library support
4. List some commercial Mobile operating systems.

- Windows mobile
- Palm OS
- Symbian OS
- iOS
- Android
- Blackberry OS

5. List the Android software stack.

- Application layer
- Application framework
- Libraries and runtime
- Kernel

6. What are the Android application components?

- Activity
- Content providers
- Service
- Broadcast receiver

7. What are the advantages of Android?

- The mobile platform Android is an open platform and can be ported on almost every type of cell phone.
- The android SDK to develop application is possible on every operating system
- Android requires a low footprint of 250kb.

8. Define M-commerce.

Mobile commerce is simple word, involves carrying out any activity related to buying and selling of commodities, services or information using the mobile hand-held devices.

9. List the types of application in M-Commerce.

- Business to Consumer applications (B2C).
- Business to Business applications (B2B).

10. What are B2C applications?

It is form of commerce in which products or services are sold by a business firm to a consumer.

Examples: Advertising, Mobile ticketing, Comparison shopping, Information about product etc.,.

11. What are B2B applications?

It is form of commerce in which products or services are sold by a company to its dealers.

Examples: Ordering and delivery confirmation, Stock tracking and control, Mobile inventory management etc.,.

12. List the advantages of M-Commerce.

- For the business organization, the benefits of using M-Commerce includes customer convenience, cost saving and new business opportunities
- From the customer perspective, M-Commerce provides the flexibility of anytime, anywhere shopping using just a lightweight device.

13. List the disadvantages of M-Commerce.

- Mobile device do not generally offer graphics or processing power of a PC. The users are therefore constrained to use small screen and keyboard and low resolution pictures and videos.
- The small screen of mobile device limits the complexity of applications.

14. Define Mobile Payment.

It is defined as any payment instrument where a mobile device is used to initiate, authorize and confirm an exchange of financial value in return of goods and services.

15. List the various mobile payment schemes.

- Bank account based
- Credit card based
- Micropayment

16. State the types of kernel and write its function.

- Monolithic Kernel-the kernel essentially constitutes the entire operating system code, except for the code for the shell.
- Microkernel-This approach tries to minimize the size of the kernel code.

17. Define IDE.

An integrated development environment (IDE) is a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder. The IDE may be a standalone application or may be included as part of one or more existing and compatible applications.

18. State the important features of SDK.

- Client program-runs on the developer's machine
- Daemon programs-runs on the background process
- Server program-runs on the background process.

19. Specify the motivation of Monolithic Kernel OS design.

- Kernel  
contains the entire OS operations except shell code
- Motivation  
OS services can run more securely and efficiently in supervisor mode

20. Mention the examples of Monolithic Kernel OS design.

- Windows
- Unix

21. List the advantages of Monolithic Kernel OS design.

- Provides good performance
- Always runs in supervisor mode
- More efficient and secure

22. List the disadvantages of Monolithic Kernel OS design.

Makes kernel

- Massive
- Non-modular
- Hard to tailor
- Maintain
- Extend
- Configure

23. List the advantages of Microkernel OS design.

- Flexible
- Modular Easier to port
- Easy to extend and implement

24. List the disadvantages of Microkernel OS design.

- Difficult to debug compared to application programs
- Bog in the kernel crashes the system and the debugger
- Non- reliable

25. What is Mobile OS?

- Facilitate third party development of application software
- Allow manufacturers of different brands of mobile devices to build their choice set of functionalities for the users

## **PART B**

1. Explain in detail about the Special constraints and requirements of Mobile OS.

2. Explain about the Commercial mobile operating systems.

3. What is M-Commerce? Explain about the types of M-Commerce applications.

4. Explain in detail about the structure of M-Commerce with suitable diagram. Give two advantages and disadvantages.

5. Explain about the various mobile payment systems.

6. Describe briefly the architecture of the Andriod operating system. Briefly identify the possible reasons as to why it has been able to rapidly improve its market share compared to its peers since its introduction few years ago.

7. Describe microkernel operating system. Why is microkernel based design being preferred for developing a mobile OS?

# CS6660 COMPILER DESIGN

## UNIT I INTRODUCTION TO COMPILER

### PART A

#### **1. What is a Compiler?**

A Compiler is a program that reads a program written in one language-the source language-and translates it in to an equivalent program in another language-the target language. As an important part of this translation process, the compiler reports to its user the presence of errors in the source program.

#### **2. State some software tools that manipulate source program.**

- Structure editors
- Pretty printers
- Static checkers
- Interpreters.

#### **3. Enlist the cousins of compiler.**

The following are the cousins  
of

- Preprocessors
- Assemblers
- Loaders
- Link editors.

#### **4. What are the main two parts of compilation? What are they performing?**

The two main parts are

- Analysis part breaks up the source program into constituent pieces and creates an intermediate representation of the source program.
- Synthesis part constructs the desired target program from the intermediate representation

#### **5. What is a Structure editor?**

A structure editor takes as input a sequence of commands to build a source program .The structure editor not only performs the text creation and modification functions of an ordinary text editor but it also analyzes the program text putting an appropriate hierarchical structure on the source program.

#### **6. What are a Pretty Printer and Static Checker?**

- A Pretty printer analyses a program and prints it in such a way that the structure of the program becomes clearly visible.
- A static checker reads a program, analyses it and attempts to discover potential bugs with out running the program.

#### **7. How many phases does analysis consists?**

Analysis consists of three phases

- Linear analysis
- Hierarchical analysis
- Semantic analysis

#### **8. What happens in linear analysis?**

This is the phase in which the stream of characters making up the source program is

read from left to right and grouped in to tokens that are sequences of characters having collective meaning.

### 9. What happens in Hierarchical analysis?

This is the phase in which characters or tokens are grouped hierarchically in to nested collections with collective meaning.

### 10. What happens in Semantic analysis?

This is the phase in which certain checks are performed to ensure that the components of a program fit together meaningfully.

### 11. State some compiler construction tools?

- Parse generator
- Scanner generators
- Syntax-directed translation engines
- Automatic code generator
- Data flow engines.

### 12. What is a Loader? What does the loading process do?

A Loader is a program that performs the two functions

- Loading
- Link editing

The process of loading consists of taking relocatable machine code, altering the relocatable address and placing the altered instructions and data in memory at the proper locations.

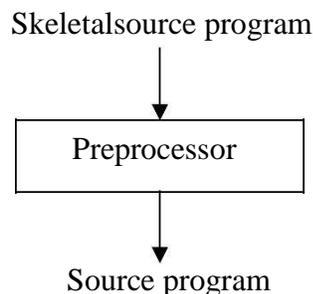
### 13. What does the Link Editing does?

**Link editing:** This allows us to make a single program from several files of relocatable machine code. These files may have been the result of several compilations, and one or more may be library files of routines provided by the system and available to any program that needs them.

### 14. What is a preprocessor?

A preprocessor is one, which produces input to compilers. A source program may be divided into modules stored in separate files. The task of collecting the source program is sometimes entrusted to a distinct program called a preprocessor.

The preprocessor may also expand macros into source language statements.



### 15. List some functions of Preprocessors.

- Macro processing
- File inclusion

- Relational Preprocessors
- Language extensions

### **16. State Symbol table.**

A Symbol table is a data structure containing a record for each identifier, with fields for the attributes of the identifier. The data structure allows us to find the record for each identifier quickly and to store or retrieve data from that record quickly.

### **17. State the general phases of a compiler.**

- Lexical analysis
- Syntax analysis
- Semantic analysis
- Intermediate code generation
- Code optimization
- Code generation

### **18. What is an assembler?**

Assembler is a program, which converts the source language in to assembly language.

### **19. What is the need for separating the analysis phase into lexical analysis and parsing? (Or)**

#### **What are the issues of lexical analyzer?**

- Simpler design is perhaps the most important consideration. The separation of lexical analysis from syntax analysis often allows us to simplify one or the other of these phases.
- Compiler efficiency is improved.
- Compiler portability is enhanced.

### **20. What is Lexical Analysis?**

The first phase of compiler is Lexical Analysis. This is also known as linear analysis in which the stream of characters making up the source program is read from left-to-right and grouped into tokens that are sequences of characters having a collective meaning.

### **21. What is a lexeme? Define a regular set.**

- A Lexeme is a sequence of characters in the source program that is matched by the pattern for a token.
- A language denoted by a regular expression is said to be a regular set

### **22. Enlist Programming Language basics.**

- The static/Dynamic distinction
- Environments and states
- Static scope and block structures
- Explicit access control
- Dynamic scope
- Parameter passing mechanisms
- Aliasing

## **PART B**

1. Discuss the phases of Compiler.
2. Describe compiler construction tools.
3. Explain Programming Language basics.
4. Explicate the errors encountered in different phases.
5. Illustrate the grouping of phases.

## UNIT-II LEXICAL ANALYSIS

### PART A

#### **1. State Lexical Analysis.**

The first phase of compiler is Lexical Analysis. This is also known as linear analysis in which the stream of characters making up the source program is read from left-to-right and grouped into tokens that are sequences of characters having a collective meaning.

#### **2. What is a lexeme? Define a regular set.**

A Lexeme is a sequence of characters in the source program that is matched by the pattern for a token.

A language denoted by a regular expression is said to be a regular set.

#### **3. What is a sentinel? What is its usage?**

A Sentinel is a special character that cannot be part of the source program.

Normally we use 'eof' as the sentinel. This is used for speeding-up the lexical analyzer.

#### **4. State the rules, which define regular expression?**

Regular expression is a method to describe regular language

If  $a$  is a symbol in  $\Sigma$ , then  $a$  is a regular expression that denotes  $\{a\}$

Suppose  $r$  and  $s$  are regular expressions denoting the languages  $L(r)$  and  $L(s)$  Then,

- $(r)/(s)$  is a regular expression denoting  $L(r) \cup L(s)$ .
- $(r)(s)$  is a regular expression denoting  $L(r)L(s)$
- $(r)^*$  is a regular expression denoting  $L(r)^*$ .
- $(r)$  is a regular expression denoting  $L(r)$ .

#### **5. What are the Error-recovery actions in a lexical analyzer?**

Deleting an extraneous character

Inserting a missing character

Replacing an incorrect character by a correct character

Transposing two adjacent characters

#### **6. Construct Regular expression for the language**

$L = \{w \in \{a,b\}^* / w \text{ ends in } abb\}$

Ans:  $\{a/b\}^*abb$ .

#### **7. State recognizer?**

Recognizers are machines. These are the machines which accept the strings belonging to certain language. If the valid strings of such language are accepted by the machine then it is said that the corresponding language is accepted by that machine, otherwise it is rejected.

#### **8. Differentiate compiler and interpreter.**

Compiler produces a target program whereas an interpreter performs the operations implied by the source program.

#### **9. Write short notes on buffer pair.**

Concerns with efficiency issues Used with a lookahead on the input

It is a specialized buffering technique used to reduce the overhead required to process an input character. Buffer is divided into two N-character halves. Use two pointers.

Used at times when the lexical analyzer needs to look ahead several characters beyond the lexeme for a pattern before a match is announced.

#### **10. Differentiate tokens, patterns, lexeme.**

**Tokens-** Sequence of characters that have a collective meaning.

**Patterns-** There is a set of strings in the input for which the same token is produced as output. This set of strings is described by a rule called a pattern associated with the token

**Lexeme-** A sequence of characters in the source program that is matched by the pattern for a token.

**11. List the operations on languages.**

**Union** -  $L \cup M = \{s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$

**Concatenation** -  $LM = \{st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$

**Kleene Closure** -  $L^*$  (zero or more concatenations of L)

**Positive Closure** -  $L^+$  (one or more concatenations of L)

**12. Write a regular expression for an identifier.**

An identifier is defined as a letter followed by zero or more letters or digits. The regular expression for an identifier is given as **letter (letter | digit)\***

**13. Mention the various notational shorthands for representing regular expressions.**

One or more instances (+)

Zero or one instance (?) \

Character classes ([abc] where a,b,c are alphabet symbols denotes the regular expressions a | b | c.)

Non regular sets

**14. What is the function of a hierarchical analysis?**

Hierarchical analysis is one in which the tokens are grouped hierarchically into nested collections with collective meaning. Also termed as Parsing.

**15. What does a semantic analysis do?**

Semantic analysis is one in which certain checks are performed to ensure that components of a program fit together meaningfully. Mainly performs type checking.

**16. Differentiate NFA and DFA.**

DFA(Deterministic Finite Automata) is the one which has a distinguished state transition for every input that it encounters	NFA(Non-Deterministic Finite Automata) is the one which has either more than one or no state transitions at all for an input that it encounters
DFA's do not have $\epsilon$ transitions	NFA's have $\epsilon$ transitions
DFA's do not have contradictory transitions	NFA's can have contradictory transitions
As every DFA is an NFA, DFA's do not need to	NFA's can be converted to equivalent DFA's using

be specifically converted to equivalent NFA's.	$\epsilon$ -closure
DFA's are relatively more difficult to construct than NFA's	NFA's are easier to construct

### 17. Define DFA.

DFA Formal Definition (reminder) A deterministic finite automaton (DFA) is a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$ ,

Where

$Q$  is a finite set called the states,

$\Sigma$  is a finite set called the alphabet,

$\delta : Q \times \Sigma \rightarrow Q$  is the transition function,

$q_0 \in Q$  is the start state, and

$F \subseteq Q$  is the set of accept states

### 18. Define automaton.

An automaton with a finite number of states is called a **Finite Automaton**(FA) or **Finite State Machine** (FSM).

#### Formal definition of a Finite Automaton

An automaton can be represented by a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$ , where –

- $Q$  is a finite set of states.
- $\Sigma$  is a finite set of symbols, called the **alphabet** of the automaton.
- $\delta$  is the transition function.
- $q_0$  is the initial state from where any input is processed ( $q_0 \in Q$ ).
- $F$  is a set of final state/states of  $Q$  ( $F \subseteq Q$ ).

19. Write short notes on alphabet.

- **Definition** – An **alphabet** is any finite set of symbols.
- **Example** –  $\Sigma = \{a, b, c, d\}$  is an **alphabet set** where 'a', 'b', 'c', and 'd' are **symbols**.

20. Define String.

- **Definition** – A **string** is a finite sequence of symbols taken from  $\Sigma$ .
- **Example** – 'cabcad' is a valid string on the alphabet set  $\Sigma = \{a, b, c, d\}$

21. How to calculate Length of a String?

- **Definition** – It is the number of symbols present in a string. (Denoted by  $|S|$ ).

- **Examples** –

- If  $S = \text{'cabcad'}$ ,  $|S| = 6$
- If  $|S| = 0$ , it is called an **empty string** (Denoted by  $\lambda$  or  $\epsilon$ )

22. What is Kleene Star?

- **Definition** – The Kleene star,  $\Sigma^*$ , is a unary operator on a set of symbols or strings,  $\Sigma$ , that gives the infinite set of all possible strings of all possible lengths over  $\Sigma$  including  $\lambda$ .

- **Representation** –  $\Sigma^* = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \dots$  where  $\Sigma_p$  is the set of all possible strings of length  $p$ .

- **Example** – If  $\Sigma = \{a, b\}$ ,  $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, \dots\}$

23. What is Kleene Closure / Plus?

- **Definition** – The set  $\Sigma^+$  is the infinite set of all possible strings of all possible lengths over  $\Sigma$  excluding  $\lambda$ .

- **Representation** –  $\Sigma^+ = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \cup \dots$

$$\Sigma^+ = \Sigma^* - \{ \lambda \}$$

- **Example** – If  $\Sigma = \{a, b\}$ ,  $\Sigma^+ = \{a, b, aa, ab, ba, bb, \dots\}$

24. Define Language.

- **Definition** – A language is a subset of  $\Sigma^*$  for some alphabet  $\Sigma$ . It can be finite or infinite.

- **Example** – If the language takes all possible strings of length 2 over  $\Sigma = \{a, b\}$ , then  $L = \{ab, bb, ba, aa\}$

## PART B

1. Discuss roles and tasks of a lexical analyzer?
2. Explain the Conversion of Regular Expression into a Deterministic Finite Automaton.
3. Describe Minimization of DFA.
4. Explain the Language for Specifying Lexical Analyzers.
5. Design of Lexical Analyzer for a sample Language.

## UNIT III SYNTAX ANALYSIS

### PART A

#### **1. Define parser.**

Hierarchical analysis is one in which the tokens are grouped hierarchically into nested collections with collective meaning.

#### **2. Mention the basic issues in parsing.**

There are two important issues in parsing.

Specification of syntax

Representation of input after parsing.

#### **3. Why lexical and syntax analyzers are separated out?**

Reasons for separating the analysis phase into lexical and syntax analyzers:

Simpler design.

Compiler efficiency is improved.

Compiler portability is enhanced.

#### **4. Define a context free grammar.**

A context free grammar  $G$  is a collection of the following

$V$  is a set of non terminals

$T$  is a set of terminals

$S$  is a start symbol

$P$  is a set of production rules

$G$  can be represented as  $G = (V, T, S, P)$

Production rules are given in the following

form

Non terminal  $\rightarrow (V \cup T)^*$

#### **5. Briefly explain the concept of derivation.**

Derivation from  $S$  means generation of string  $w$  from  $S$ . For constructing derivation two things are important.

Choice of non terminal from several others.

Choice of rule from production rules for corresponding non terminal. Instead of choosing the arbitrary non terminal one can choose

either leftmost derivation – leftmost non terminal in a sentinel form

or rightmost derivation – rightmost non terminal in a sentinel form

#### **6. Define ambiguous grammar.**

A grammar  $G$  is said to be ambiguous if it generates more than one parse tree for some sentence of language  $L(G)$ .

i.e. both leftmost and rightmost derivations are same for the given sentence.

#### **7. What is an operator precedence parser?**

A grammar is said to be operator precedence if it possesses the following properties:

No production on the right side is  $\epsilon$ .

There should not be any production rule possessing two adjacent non terminals at the right hand side.

### 8. List the properties of LR parser.

LR parsers can be constructed to recognize most of the programming languages for which the context free grammar can be written.

The class of grammar that can be parsed by LR parser is a superset of class of grammars that can be parsed using predictive parsers.

LR parsers work using non backtracking shift reduce technique yet it is efficient one.

### 9. Mention the types of LR parser.

SLR parser- simple LR parser

LALR parser- lookahead LR parser

Canonical LR parser

### 10. What are the problems with top down parsing?

The following are the problems associated with top down parsing:

Backtracking

Left recursion

Left factoring

Ambiguity

### 11. Write the algorithm for FIRST and FOLLOW.

#### FIRST

If X is terminal, then  $FIRST(X)$  IS  $\{X\}$ .

If  $X \rightarrow \epsilon$  is a production, then add  $\epsilon$  to  $FIRST(X)$ .

If X is non terminal and  $X \rightarrow Y_1 Y_2 \dots Y_k$  is a production, then place a in  $FIRST(X)$  if for some i, a is in  $FIRST(Y_i)$ , and  $\epsilon$  is in all of  $FIRST(Y_1), \dots, FIRST(Y_{i-1})$ ;

#### FOLLOW

1. Place \$ in  $FOLLOW(S)$ , where S is the start symbol and \$ is the input right endmarker.

2. If there is a production  $A \rightarrow \alpha B \beta$ , then everything in  $FIRST(\beta)$  except for  $\epsilon$  is placed in  $FOLLOW(B)$ .

3. If there is a production  $A \rightarrow \alpha B$ , or a production  $A \rightarrow \alpha B \beta$  where  $FIRST(\beta)$  contains  $\epsilon$ , then everything in  $FOLLOW(A)$  is in  $FOLLOW(B)$ .

### 12. List the advantages and disadvantages of operator precedence parsing.

#### Advantages

This type of parsing is simple to implement.

#### Disadvantages

1. The operator like minus has two different precedence (unary and binary). Hence it is hard to handle tokens like minus sign.

2. This kind of parsing is applicable to only small class of grammars.

### 13. What is dangling else problem?

Ambiguity can be eliminated by means of dangling-else grammar which is show below:

stmt  $\rightarrow$  if expr then stmt

| if expr then stmt else stmt

| other

### 14. Write short notes on YACC.

YACC is an automatic tool for generating the parser program.

YACC stands for Yet Another Compiler Compiler which is basically the utility available from UNIX.

Basically YACC is LALR parser generator.

It can report conflict or ambiguities in the form of error messages.

### 15. What is meant by handle pruning?

A rightmost derivation in reverse can be obtained by handle pruning.

If  $w$  is a sentence of the grammar at hand, then  $w = \gamma_n$ , where  $\gamma_n$  is the  $n$ th right-sentential form of some as yet unknown rightmost derivation

$S = \gamma_0 \Rightarrow \gamma_1 \dots \Rightarrow \gamma_{n-1} \Rightarrow \gamma_n = w$

### 16. Define LR(0) items.

An LR(0) item of a grammar  $G$  is a production of  $G$  with a dot at some position of the right side. Thus, production  $A \rightarrow XYZ$  yields the four items

$A \rightarrow \cdot XYZ$

$A \rightarrow X \cdot Y$

$Z$

$A \rightarrow XY \cdot$

$Z$

$A \rightarrow XYZ$

.

### 17. What is meant by viable prefixes?

The set of prefixes of right sentential forms that can appear on the stack of a shift-reduce parser are called viable prefixes. An equivalent definition of a viable prefix is that it is a prefix of a right sentential form that does not continue past the right end of the rightmost handle of that sentential form.

### 18. Define handle.

A handle of a string is a substring that matches the right side of a production, and whose reduction to the nonterminal on the left side of the production represents one step

along the reverse of a rightmost derivation.

A handle of a right – sentential form  $\gamma$  is a production  $A \rightarrow \beta$  and a position of  $\gamma$  where the string  $\beta$  may be found and replaced by  $A$  to produce the previous right-sentential form in a rightmost derivation of  $\gamma$ . That is, if  $S \Rightarrow \alpha A w \Rightarrow \alpha \beta w$ , then  $A \rightarrow \beta$  in the position following  $\alpha$  is a handle of  $\alpha \beta w$ .

### 19. What are kernel & non-kernel items?

**Kernel items**, which include the initial item,  $S' \rightarrow .S$ , and all items whose dots are not at the left end.

**Non-kernel items**, which have their dots at the left end.

### 20. What is phrase level error recovery?

Phrase level error recovery is implemented by filling in the blank entries in the predictive parsing table with pointers to error routines. These routines may change, insert, or delete symbols on the input and issue appropriate error messages. They may also pop from the stack.

## PART B

1. Construct a predictive parsing table for the grammar

$E \rightarrow E + T / F$

$T \rightarrow T * F /$

$F$

$F \rightarrow (E) / id$

2. Give the LALR parsing table for the grammar.

$S \rightarrow L = R / R$

$L \rightarrow * R / id$

$R \rightarrow L$

3. Consider the grammar

$E \rightarrow TE'$

$E' \rightarrow + TE' / E$

$T \rightarrow FT'$

$T' \rightarrow *FT' / E$

$F \rightarrow (E) / id$

Construct a predictive parsing table for the grammar shown above. Verify whether the input string  $id + id * id$  is accepted by the grammar or not. 4) Consider the grammar.

$E \rightarrow E +$

T  
 E -> T  
 T -> T \*  
 F  
 T -> F  
 F -> (E) / id

Construct an LR parsing table for the above grammar. Give the moves of the LR parser  
 On id \* id + id

4. Compare top down parsing and bottom up parsing methods.
5. What are LR parsers? Explain with a diagram the LR parsing algorithm.
6. Explain recursive descent parser with appropriate examples.
7. Compare SLR, LALR and LR parses.

## UNIT IV - SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT

### PART A

#### **1. List the different storage allocation strategies.**

The strategies are:

- Static allocation
- Stack allocation
- Heap allocation

#### **2. What are the contents of activation record?**

The activation record is a block of memory used for managing the information needed by a single execution of a procedure. Various fields of activation record are:

- Temporary variables
- Local variables
- Saved machine registers
- Control link
- Access link
- Actual parameters
- Return values

#### **3. What is dynamic scoping?**

In dynamic scoping a use of non-local variable refers to the non-local data declared in most recently called and still active procedure. Therefore each time new findings are set up for local names called procedure. In dynamic scoping symbol tables can be required at run time.

#### **4. Classify attributes.**

- Synthesized attributes
- Inherited attributes.

#### **5. What is code motion?**

Code motion is an optimization technique in which amount of code in a loop is decreased. This transformation is applicable to the expression that yields the same result independent of the number of times the loop is executed. Such an expression is placed before the loop.

## 6. What are the properties of optimizing compiler?

The source code should be such that it should produce minimum amount of target code.

There should not be any unreachable code.

Dead code should be completely removed from source language.

The optimizing compilers should apply following code improving transformations on source language.

Common sub expression elimination

Dead code elimination

Code movement

Strength reduction

## 7. What are the various ways to pass a parameter in a function?

Call by value

Call by reference

Copy-restore

Call by name

## 8. What is SDD?

Syntax directed definition specifies the values of attributes by associating semantic rules with the grammar productions. It is a context free grammar with attributes and rules together which are associated with grammar symbols and productions respectively.

The process of syntax directed translation is two-fold:

- Construction of syntax tree and Computing values of attributes at each node by visiting the nodes of syntax tree.

## 9. Define annotated parse tree?

It is a parse tree showing the values of the attributes at each node. The process of computing the attribute values at the nodes is called annotating or decorating the parse tree.

## 10. What are the types of translation?

### L-attributed translation

It performs translation during parsing itself.

No need of explicit tree construction.

L represents 'left to right'.

### S-attributed translation

It is performed in connection with bottom up parsing.

'S' represents synthesized.

## 11. What are the functions for constructing syntax trees for expressions?

The construction of a syntax tree for an expression is similar to the translation of the expression into postfix form.

Each node in a syntax tree can be implemented as a record with several fields.

## 12. Give short note about call-by-name?

Call by name, at every reference to a formal parameter in a procedure body the name of the corresponding actual parameter is evaluated. Access is then made to the effective parameter.

### 13. How parameters are passed to procedures in call-by-value method?

This mechanism transmits values of the parameters of call to the called program. The transfer is one way only and therefore the only way to returned can be the value of a function.

```
Main ( )
{ print (5); }
Int
Void print (int n)
{ printf (“%d”, n); }
```

### 14. Define static allocations and stack allocations.

Static allocation is defined as lays out for all data objects at compile time.

Names are bound to storage as a program is compiled, so there is no need for a run time support package.

Stack allocation is defined as process in which manages the run time as a Stack. It is based on the idea of a control stack; storage is organized as a stack, and activation records are pushed and popped as activations begin and end.

### 15. Enlist the units of activation record.

Temporaries	Stores temporary and intermediate values of an expression.
Local Data	Stores local data of the called procedure.
Machine Status	Stores machine status such as Registers, Program Counter etc., before the procedure is called.
Control Link	Stores the address of activation record of the caller procedure.
Access Link	Stores the information of data which is outside the local scope.
Actual Parameters	Stores actual parameters, i.e., parameters which are used to send input to the called procedure.
Return Value	Stores return values.

### 16. Define formal parameters and actual parameters.

#### Formal Parameters

Variables that take the information passed by the caller procedure are called formal parameters.

These variables are declared in the definition of the called function.

#### Actual Parameters

Variables whose values or addresses are being passed to the called procedure are called actual parameters. These variables are specified in the function call as arguments.

### **17. Define a syntax-directed translation?**

Syntax-directed translation specifies the translation of a construct in terms of Attributes associated with its syntactic components. Syntax-directed translation uses a context free grammar to specify the syntactic structure of the input. It is an input-output mapping.

### **18. Define an attribute. Give the types of an attribute?**

An attribute may represent any quantity, with each grammar symbol, it associates a set of attributes and with each production, a set of semantic rules for computing values of the attributes associated with the symbols appearing in that production.

**Example:** a type, a value, a memory location etc.,  
Synthesized attributes.  
Inherited attributes.

### **19. Give the 2 attributes of syntax directed translation into 3-addr code?**

E.place, the name that will hold the value of E and  
E.code , the sequence of 3-addr statements evaluating E.

### **20. Write the grammar for flow-of-control statements?**

The following grammar generates the flow-of-control statements, if-then, if-then-else, and while-do statements.

```
S -> if E
    then S1
    | If E then S1
    else S2
    | While E do S1.
```

## **PART B**

1. Discuss in detail about the run time storage arrangement.
2. What are different storage allocation strategies? Explain.
3. Write in detail about the issues in the design of code generator.
4. Explain how declarations are done in a procedure using syntax directed translations.
5. What is a three address code? Mention its types. How would you implement these address statements? Explain with suitable examples.

## **UNIT V CODE OPTIMIZATION AND CODE GENERATION**

### **PART A**

#### **1. Define code generations with ex?**

It is the final phase in compiler model and it takes as an input an intermediate representation of the source program and output produces as equivalent target programs. Then intermediate instructions are each translated into a sequence of machine instructions that perform the same task.

#### **2. What are the issues in the design of code generator?**

Input to the generator  
Target programs

Memory management  
 Instruction selection  
 Register allocation  
 Choice of evaluation order  
 Approaches to code generation.

**3. Give the variety of forms in target program.**

Absolute machine language.  
 Relocatable machine language.  
 Assembly language.

**4. Give the factors of instruction selections.**

Uniformity and completeness of the instruction sets  
 Instruction speed and machine idioms  
 Size of the instruction sets.

**5. What are the sub problems in register allocation strategies?**

During register allocation, we select the set of variables that will reside in register at a point in the program.  
 During a subsequent register assignment phase, we pick the specific register that a variable reside in.

**6. Mention the properties that a code generator should possess.**

The code generator should produce the correct and high quality code. In other words, the code generated should be such that it should make effective use of the resources of the target machine. Code generator should run efficiently

**7. List the terminologies used in basic blocks.**

**Define and use** – the three address statement  $a:=b+c$  is said to define a and to use b and c.

**Live and dead** – the name in the basic block is said to be live at a given point if its value is used after that point in the program. And the name in the basic block is said to be dead

**8. Write the addressing mode and associated costs in the target machine.**

MODE	FORM	ADDRESS	ADDED COST
Absolute	M	M	1
Register	R	R	0
Indexed	c(R)	c+contents(R)	1
Indirect register	*R	contents(R)	0
Indirect indexed	*c(R)	contents(c+contents(R))	1

## 9. Define basic block and flow graph.

A basic block is a sequence of consecutive statements in which flow of Control enters at the beginning and leaves at the end without halt or possibility Of branching except at the end.

A flow graph is defined as the adding of flow of control information to the Set of basic blocks making up a program by constructing a directed graph.

## 10. Write the step to partition a sequence of 3 address statements into basic blocks.

1. First determine the set of leaders, the first statement of basic blocks.

The rules we can use are the following. The first statement is a leader.

Any statement that is the target of a conditional or unconditional goto is a leader.

Any statement that immediately follows a goto or conditional goto statement is a leader.

2. For each leader, its basic blocks consists of the leader and all statements Up to but not including the next leader or the end of the program.

## 11. Give the important classes of local transformations on basic blocks

Structure preservation transformations

Algebraic transformations.

## 12. Describe algebraic transformations.

It can be used to change the set of expressions computed by a basic blocks into A algebraically equivalent sets.

The useful ones are those that simplify the expressions place expensive operations by cheaper ones.

$$X = X + 0$$

$$X = X * 1$$

## 13. What is meant by register descriptors and address descriptors?

A register descriptor keeps track of what is currently in each register. It is Consulted whenever a new register is needed.

An address descriptor keeps track of the location where ever the current Value of the name can be found at run time.

## 14. What are the actions to perform the code generation algorithms?

Invoke a function get reg to determine the location L.

Consult the address descriptor for y to determine  $y^c$ , the current location of y.

If the current values of y and/or z have no next uses, are not live on exit from the block, and are in register, alter the register descriptor.

## 15. Write the code sequence for the $d := (a-b) + (a-c) + (a-c)$ .

Statement	Code generation	Register descriptor	Address
			descriptor
t:=a-b	MOV a,R0 SUB b,R0	R0 contains t	t in R0
u:=a-c	MOV a,R1 SUB c,R1	R0 contains t R1 contains u	t in R0 u in R1
v:=t+u	ADD R1,R0	R0 contains v R1 contains u	u in R1 v in R0
d:=v+u	ADD R1,R0 MOV R0,d	R0 contains d	d in R0 d in R0 and memory

**16. Write the labels on nodes in DAG.**

A DAG for a basic block is a directed acyclic graph with the following Labels on nodes:  
 Leaves are labeled by unique identifiers, either variable names or constants.  
 Interior nodes are labeled by an operator symbol.  
 Nodes are also optionally given a sequence of identifiers for labels.

**17. Give the applications of DAG.**

Automatically detect the common sub expressions  
 Determine which identifiers have their values used in the block.  
 Determine which statements compute values that could be used outside the blocks.

**18. How will you find the leaders in the basic block?**

The rules used to find the leaders in the basic block are the following:  
 The first statement is a leader.  
 Any statement that is the target of a conditional or unconditional goto is a leader.  
 Any statement that immediately follows a goto or conditional goto statement is a leader.

**19. Write the characteristics of peephole optimization.**

Redundant-instruction elimination  
 Flow-of-control optimizations.  
 Algebraic simplifications  
 Use of machine idioms

**20. What are the structure preserving transformations on basic blocks?**

Common sub-expression elimination  
 Dead-code elimination  
 Renaming of temporary variables  
 Interchange of two independent adjacent statement

## 21. Define Common sub-expression elimination with ex.

It is defined as the process in which eliminate the statements which has the Same expressions. Hence this basic block may be transformed into the equivalent Block.

**Ex:**

```
a := b +
c
b := a - d
c := b + c
```

**After elimination:**

```
a := b +
c
b := a - d
c := a
```

## 22. Define Dead-code elimination with ex.

It is defined as the process in which the statement  $x=y+z$  appear in a basic block, where  $x$  is a dead that is never subsequently used. Then this statement maybe safely removed without changing the value of basic blocks.

## 23. Define Renaming of temporary variables with ex.

We have the statement  $u:=b + c$  ,where  $u$  is a new temporary variable, and change all uses of this instance of  $t$  to  $u$ , then the value of the basic block is not changed.

## 24. Define reduction in strength with ex.

Reduction in strength replaces expensive operations by equivalent cheaper ones on the target machines. Certain machine instructions are cheaper than others and can often be used as special cases of more expensive operators.

**Ex:**

$X^2$  is invariably cheaper to implement as  $x*x$  than as a call to an exponentiation routine.

## 25. Define use of machine idioms.

The target machine may have harder instructions to implement certain specific operations efficiently. Detecting situations that permit the use of these instructions can reduce execution time significantly.

## 26. Define code optimization and optimizing compiler

The **term code-optimization** refers to techniques a compiler can employ in an attempt to produce a better object language program than the most obvious for a given source program. Compilers that apply code-improving transformations are called **Optimizing-compilers**.

### PART B:

1. What are the issues in the design of code generator? Explain in detail.
2. Discuss about the run time storage management.

3. Explain basic blocks and flow graphs.
4. Explain about transformation on a basic block.
5. Write a code generation algorithm. Explain about the descriptor and function getreg(). Give an example.
6. Explain peephole optimization .
7. Explain DAG representation of basic blocks.
8. Explain principle sources of code optimization in details.
9. Explain the Source language issues with details.
10. Explain the Storage organization strategies with examples.

## **CS6659 ARTIFICIAL INTELLIGENCE**

### **PART - A**

#### **UNIT I INTRODUCTION TO AI AND PRODUCTION SYSTEMS**

1. Define an agent.

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon the environment through effectors.

2. Define rational agent.

A rational agent is one that does the right thing. Here right thing is one that will cause agent to be more successful. That leaves us with the problem of deciding how and when to evaluate the agent's success.

3. Define an agent program.

Agent program is a function that implements the agents mapping from percept to actions.

4. List the various type of agent program.

- Simple reflex agent program.
- Agent that keep track of the world.
- Goal based agent program.
- Utility based agent program

5. State the various properties of environment.

Accessible Vs Inaccessible:

If an agent's sensing apparatus give it access to the complete state of the environment then we can say the environment is accessible to he agent.

Deterministic Vs Non deterministic:

If the next state of the environment is completely determined by the current state and the actions selected by the agent, then the environment is deterministic.

Episodic Vs Non episodic:

In this, agent's experience is divided into episodes. Each episode consists of agents perceiving and then acting. The quality of the action depends on the episode itself because subsequent episodes do not depend on what actions occur in previous experiences.

Discrete Vs Continuous:

If there is a limited number of distinct clearly defined percepts & actions we say that the environment is discrete.

6. What are the phases involved in designing a problem solving agent?

The three phases are: Problem formulation, Search solution, Execution.

7. What are the different types of problem?

Single state problem, Multiple state problem, Contingency problem, Exploration problem.

8. Define problem.

A problem is really a collection of information that the agent will use to decide what to do.

9. List the basic elements that are to be included in problem definition.

Initial state, operator, successor function, state space, path, goal test, path cost.

10. Mention the criteria for the evaluation of search strategy.

There are 4 criteria: Completeness, time complexity, space complexity, optimality.

11. Differentiate blind search & heuristic search.

Blind search has no information about the number of steps or the path cost from the current state to the goal, they can distinguish a goal state from a non-goal state. Heuristic search—knowledge given. Problem specification solution is best.

12. List the various search strategies.

i) uninformed search strategies

- a. BFS
- b. Uniform cost search
- c. DFS
- d. Depth limited search
- e. Iterative deepening search
- f. Bidirectional search

ii) Informed search strategies

- Greedy search.
- A\* search.

13. Whether uniform cost search is optimal?

Uniform cost search is optimal & it chooses the best solution depending on the path cost.

14. Write the time & space complexity associated with depth limited search.

Time complexity =  $O(b^d)$ , b-branching factor, d-depth of tree

Space complexity =  $O(b)$

15. Define CSP

A constraint satisfaction problem is a special kind of problem satisfies some additional structural properties beyond the basic requirements for problem in general. In a CSP; the states are defined by the values of a set of variables and the goal test specifies a set of constraint that the value must obey.

16. Give the drawback of DFS.

The drawback of DFS is that it can get stuck going down the wrong path. Many problems have very deep or even infinite search tree. So dfs will never be able to recover from an unlucky choice at one of the nodes near the top of the tree. So DFS should be avoided for search trees with large or infinite maximum depths.

17. List the various AI Application Areas

- natural language processing - understanding,
- generating, translating;
- planning;
- vision - scene recognition, object recognition, face
- recognition;
- robotics;
- theorem proving;
- speech recognition;
- game playing;
- problem solving ,Expert systems etc.

18. Define search tree.

The tree which is constructed for the search process over the state space is called search tree.

19. Define search node.

The root of the search tree that is the initial state of the problem is called search node.

20. Define fringe.

The collection of nodes that have been generated but not yet expanded, this collection is called fringe or frontier.

21. Define Evaluation function,  $f(n)$ .

A node with the lowest evaluation is selected for expansion, because evaluation measures distance to the goal.

22. Define Heuristic function,  $h(n)$ .

$h(n)$  is defined as the estimated cost of the cheapest path from node  $n$  to a goal node.

23. Define Greedy Best First Search.

It expands the node that is closest to the goal (i.e.) to reach solution in a quicker way. It is done by using the heuristic function:  $f(n) = h(n)$ .

24. Define A\* search.

A\* search evaluates nodes by combining  $g(n)$ , the cost to reach the node and  $h(n)$ , the cost to get from the node to the goal.

$$f(n) = g(n) + h(n)$$

## PART B

1. What is artificial intelligence?
2. Problem formulation & Problem Definition.
3. Production systems.
4. What is Search?
5. CONSTRAINT SATISFACTION PROBLEMS(CSP)

## UNIT – II REPRESENTATION OF KNOWLEDGE

### PART – A

1. What are Logical agents

Logical agents apply inference to a knowledge base to derive new information and make decisions.

2. Give an example rule for Goal Based Agent.

Once the gold is found, it is necessary to change strategies. So now we need a new set of action values. We could encode this as a rule:

$$\forall s \text{ Holding}(\text{Gold}, s) \Rightarrow \text{Goal Location}([2,3],s)$$

3. What are the components of Propositional Logic?

- Logical constants: true, false
- Propositional symbols: P, Q, S, ... (atomic sentences)
- Wrapping parentheses: ( ... )
- Sentences are combined by connectives:

$\wedge$  ...and [conjunction]

$\vee$  ...or [disjunction]

$\rightarrow$ ...implies [implication / conditional]

$\leftrightarrow$ ...is equivalent [bi conditional]

$\neg$  ...not [negation]

- Literal: atomic sentence or negated atomic sentence

#### 4. Define First Order Logic.

- First-order logic (FOL) models the world in terms of
  - Objects, which are things with individual identities
  - Properties of objects that distinguish them from other objects
  - Relations that hold among sets of objects
  - Functions, which are a subset of relations where there is only one “value” for any given “input”.
- Examples:
  - Objects: Students, lectures, companies, cars ...
  - Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - Properties: blue, oval, even, large, ...
  - Functions: father-of, best-friend, second-half, one-more-than ...

#### 5. What are the types of Quantifiers?

Universal Quantifiers & Existential Quantifiers

#### 6. What is Universal Quantification?

Universal quantification

- a.  $\forall x P(x)$  means that P holds for all values of x in the domain associated with that variable

Example.  $(\forall x) \text{dolphin}(x) \wedge \text{mammal}(x)$

#### 7. What is Existential quantification

- a.  $(\exists x)P(x)$  means that P holds for some value of x in the domain associated with that variable

Example:  $(\exists x) \text{mammal}(x) \wedge \text{lays-eggs}(x)$

#### 8. Define a knowledge Base

Knowledge base is the central component of knowledge base agent and it is described as a set of representations of facts about the world.

### 9. Define a Sentence?

Each individual representation of facts is called a sentence. The sentences are expressed in a language called as knowledge representation language.

### 10. Define an inference procedure.

An inference procedure reports whether or not a sentence ( $\alpha$ ) is entailed by knowledge base provided a knowledge base and a sentence ( $\alpha$ ). An inference procedure 'i' can be described by the sentences that it can derive. If i can derive sentence ( $\alpha$ ) from knowledge base, we can write.

KB  $\vdash$   $\alpha$  is derived from KB or i derives  $\alpha$  from KB

### 11. Define Syntax?

Syntax is the arrangement of words. Syntax of a knowledge describes the possible configurations that can constitute sentences. Syntax of the language describes how to make sentences.

### 12. Define Semantics

The semantics of the language defines the truth of each sentence with respect to each possible world. With this semantics, when a particular configuration exists with in an agent, the agent believes the corresponding sentence.

### 13. Define Logic.

Logic is one which consist of

- i. A formal system for describing states of affairs, consisting of a) Syntax b)Semantics.
- ii. Proof Theory – a set of rules for deducing the entailment of a set sentences.

### 14. What is entailment?

The relation between sentences is called entailment. The formal definition of entailment is this  $\vdash$  if and only if in every model in which is true, is also true or if is true then must also be true.

### 15. What is truth preserving?

An inference algorithm that derives only entailed sentences is called sound or truth preserving

### 16. Define a Proof

A sequence of application of inference rules is called a proof. Finding proof is exactly finding solution to search problems. If the successor function is defined to generate all possible applications of inference rules then the search algorithms can be applied to find proofs.

17. Define a Complete inference procedure

An inference procedure is complete if it can derive all true conditions from a set of premises.

18. Define Modus Ponens's rule in Propositional logic?

The standard patterns of inference that can be applied to derive chains of conclusions that lead to the desired goal is said to be Modus Ponens's rule.

19. Define Global minimum.

If elevation corresponds to cost, then the aim is to find the lowest valley is called global minimum.

20. Define Global Maximum.

If elevation corresponds to an objective function, then the aim is to find the highest peak is called global maximum.

21. What is the meaning for greedy local search?

It goals (picks) a good neighbor state without thinking ahead about where to go next.

22. Define annealing.

Annealing is the process used to harden metals (or) glass by heating them to a high temperature and then gradually cooling them, thus allowing the material to coalesce into a low energy crystalline state.

23. Define simulated annealing.

This algorithm, instead of picking the best move, it picks a random move. If the move improves the situation, it is always accepted.

24. What is Genetic Algorithms?

Genetic Algorithm is a variant of stochastic beam search in which successor states are generated by combining two parent states, rather than by modifying a single state.

25. Define Online Search agent.

Agent operates by interleaving computation and action (i.e.) first it takes an action, and then it observes the environment and computes the next action.

## **PART B**

1. What is Knowledge? and its types?
2. Explain Knowledge representation using Predicate logic:
3. Explain Predicate

4. What is Resolution?
5. Explain Knowledge representation using other logic
6. Explain Conceptual Graphs:
7. Explain Conceptual Dependency

## UNIT – III II KNOWLEDGE INFERENCE

### PART – A

1. Why does uncertainty arise ?

- Agents almost never have access to the whole truth about their environment.
- Agents cannot find a caterorial answer.
- Uncertainty can also arise because of incompleteness, incorrectness in agents understanding of properties of environment.

2. Define the term utility?

The term utility is used in the sense of \"the quality of being useful .\", utility of a state is relative to the agents, whose preferences the utility function is supposed to represent.

3. What is the need for probability theory in uncertainty ?

Probability provides the way of summarizing the uncertainty that comes from our laziness and ignorance . Probability statements do not have quite the same kind of semantics known as evidences.

4. What is the need for utility theory in uncertainty?

Utility theory says that every state has a degree of usefulness, or utility to In agent, and that the agent will prefer states with higher utility. The use utility theory to represent and reason with preferences.

5. What Is Called As Decision Theory ?

Preferences As Expressed by Utilities Are Combined with Probabilities in the General Theory of Rational Decisions Called Decision Theory. Decision Theory = Probability Theory + Utility Theory.

6. Define Prior Probability?

$p(a)$  for the Unconditional or Prior Probability Is That the Proposition A is True. It is important to remember that  $p(a)$  can only be used when there is no other information.

7. Define conditional probability?

Once the agents has obtained some evidence concerning the previously unknown propositions making up the domain conditional or posterior probabilities with the notation  $p(A/B)$  is used. This is important that  $p(A/B)$  can only be used when all be is known.

8. Define probability distribution:

If we want to have probabilities of all the possible values of a random variable probability distribution is used. Example

$P(\text{weather}) = (0.7, 0.2, 0.08, 0.02)$ . This type of notations simplifies many equations.

9. What is an atomic event?

An atomic event is an assignment of particular values to all variables, in other words, the complete specifications of the state of domain.

10. Define joint probability distribution

This completely specifies an agent's probability assignments to all propositions in the domain. The joint probability distribution  $p(x_1, x_2, \dots, x_n)$  assigns probabilities to all possible atomic events where  $X_1, X_2, \dots, X_n = \text{variables}$ .

11. Give the Bayes rule equation

We taken  $P(A \wedge B) = P(A/B) P(B)$  ----- 1

$P(A \wedge B) = P(B/A) P(A)$  ----- 2

DIVIDING BYE  $P(A)$  , We get  $P(B/A) = \frac{P(A/B) P(B)}{P(A)}$

$P(A)$

12. What is the basic task of a probabilistic inference?

The basic task is to reason in terms of prior probabilities of conjunctions, but for the most part, we will use conditional probabilities as a vehicle for probabilistic inference.

13. What are called as Poly trees?

The algorithm that works only on singly connected networks known as Poly trees. Here at most one undirected path between any two nodes is present.

14. What is called as multiple connected graph?

A multiple connected graph is one in which two nodes are connected by more than one path.

15. List the 3 basic classes of algorithms for evaluating multiply connected graphs.

- Clustering methods
- Conditioning methods
- Stochastic simulation methods

16. Define Uncertainty.

Uncertainty means that many of the simplifications that are possible with deductive inference are no longer valid.

17. What are all the various uses of a belief network?

- Making decisions based on probabilities in the network and on the agent's utilities.
- Deciding which additional evidence variables should be observed in order to gain useful

information.

- Performing sensitivity analysis to understand which aspects of the model have the greatest impact on the probabilities of the query variables (and therefore must be accurate).
- Explaining the results of probabilistic inference to the user.

18. What is meant by learning?

Learning is a goal-directed process of a system that improves the knowledge or the knowledge representation of the system by exploring experience and prior knowledge.

19. Define informational equivalence.

A transformation from one representation to another causes no loss of information; they can be constructed from each other.

20. Define computational equivalence.

The same information and the same inferences are achieved with the same amount of effort.

21. List the difference between knowledge acquisition and skill refinement.

- knowledge acquisition (example: learning physics) — learning new symbolic information coupled with the ability to apply that information in an effective manner
- skill refinement (example: riding a bicycle, playing the piano) — occurs at a subconscious level by virtue of repeated practice

22. What is meant by analogical reasoning?

Instead of using examples as foci for generalization, one can use them directly to solve new problems.

23. Define Explanation-Based Learning.

The background knowledge is sufficient to explain the hypothesis. The agent does not learn anything factually new from the instance. It extracts general rules from single examples by explaining the examples and generalizing the explanation

24. What is meant by Relevance-Based Learning?

- uses prior knowledge in the form of determinations to identify the relevant attributes
- generates a reduced hypothesis space

25. Define Knowledge-Based Inductive Learning.

Knowledge-Based Inductive Learning finds inductive hypotheses that explain set of observations with the help of background knowledge.

26. What is truth preserving?

An inference algorithm that derives only entailed sentences is called sound or truth preserving.

27. Define Inductive learning.

Learning a function from examples of its inputs and outputs is called inductive learning.

28. How the performance of inductive learning algorithms can be measured?

It is measured by their learning curve, which shows the prediction accuracy as a function of the number of observed examples.

29. List the advantages of Decision Trees

- It is one of the simplest and successful forms of learning algorithm.
- It serves as a good introduction to the area of inductive learning and is easy to implement.

30. What is the function of Decision Trees?

A decision tree takes as input an object or situation by a set of properties, and outputs a yes/no decision. Decision tree represents Boolean functions.

31. List some of the practical uses of decision tree learning.

- Designing oil platform equipment
- Learning to fly

32. Define reinforcement learning.

The task of reinforcement learning is to use rewards to learn a successful agent function.

33. Differentiate between Passive learner and Active learner.

A passive learner watches the world going by, and tries to learn the utility of being in various states. An active learner acts using the learned information, and can use its problem generator to suggest explorations of unknown portions of the environment.

34. State the design issues that affect the learning element.

- Which components of the performance element are to be improved
- What representation is used for those components
- What feedback is available
- What prior information is available

35. State the factors that play a role in the design of a learning system.

- Learning element
- Performance element
- Critic

- Problem generator

36. What is memoization?

The technique of memorization is used to speed up programs by saving the results of computation. The basic idea is to accumulate a database of input/output pairs; When the function is called, it first checks the database to see if it can avoid solving the problem from scratch.

37. Define Q-Learning.

The agent learns an action-value function giving the expected utility of taking a given action in a given state. This is called Q-Learning.

38. Differentiate between supervised learning & unsupervised learning.

Any situation in which both inputs and outputs of a component can be perceived is called supervised learning. Learning when there is no hint at all about the correct outputs is called unsupervised learning.

39. Define Bayesian learning

Bayesian learning simply calculates the probability of each hypothesis, given the data, and makes predictions on that basis. That is, the predictions are made by using all the hypotheses, weighted by their probabilities, rather than by using just a single “best” hypothesis.

### **PART B**

1. Explain Frame based System:
2. Explain Forward Chaining and Backward chaining
3. Explain Fuzzy reasoning- Certainty factors:
4. Define Bayesian theory

## **UNIT – IV PLANNING AND MACHINE LEARNING**

### **PART – A**

1. Define state space search

The most straightforward approach is to use state-space search. Because the descriptions of actions in a planning problem specify both preconditions and effects, it is possible to search in either direction either forward from the initial state or backward from the goal

2. What are the types of state space search?

Forward state space search & backward state space search

3. Define Forward state-space search

It is sometimes called progression planning, because it moves in the forward direction.

4. What are the advantages of backward state-space search?

The main advantage of backward search is that it allows us to consider only relevant actions.

#### 5. Define Partial-Order Planning

A set of actions that make up the steps of the plan. These are taken from the set of actions in the planning problem. The “empty” plan contains just the Start and Finish actions. Start has no preconditions and has as its effect all the literals in the initial state of the planning problem. Finish has no effects and has as its preconditions the goal literals of the planning problem.

#### 6. What are the advantages of Partial-Order Planning

Partial-order planning has a clear advantage in being able to decompose problems into sub problems. It also has a disadvantage in that it does not represent states directly, so it is harder to estimate how far a partial-order plan is from achieving a goal.

#### 7. What are Planning Graphs

A Planning graph consists of a sequence of levels that correspond to time steps in the plan where level 0 is the initial state. Each level contains a set of literals and a set of Actions.

#### 8. What is Conditional planning?

Also known as contingency planning, conditional planning deals with incomplete information by constructing a conditional plan that accounts for each possible situation or contingency that could arise.

#### 9. What is action plan?

The process of checking the preconditions of each action as it is executed, rather than checking the preconditions of the entire remaining plan. This is called action monitoring.

#### 10. Define planning.

Planning can be viewed as a type of problem solving in which the agent uses beliefs about actions and their consequences to search for a solution.

#### 11. What are the components that are needed for representing an action?

The components that are needed for representing an action are:

- i. Action description.
- ii. Precondition.
- iii. Effect.

#### 12. What are the components that are needed for representing a plan?

The components that are needed for representing a plan are:

- i. A set of plans steps.
- ii. A set of ordering constraints.
- iii. A set of variable binding constraints.
- iv. A set of casual link protection.

13. What are the different types of planning?

The different types of planning are as follows:

- i. Situation space planning.
- ii. Progressive planning.
- iii. Regressive planning.
- iv. Partial order planning.
- v. Fully instantiated planning.

14. Define conditional planning.

Conditional planning is a way in which the incompleteness of information is incorporated in terms of adding a conditional step, which involves if – then rules.

15. Give the classification of learning process.

The learning process can be classified as:

- i. Process which is based on coupling new information to previously acquired knowledge
  - a. Learning by analyzing differences.
  - b. Learning by managing models.
  - c. Learning by correcting mistakes.
  - d. Learning by explaining experience.
- ii. Process which is based on digging useful regularity out of data, usually called as Data base mining:
  - a. Learning by recording cases.
  - b. Learning by building identification trees.
  - c. Learning by training neural networks.

16. State Martin's law.

The law states that, “ You cannot learn anything unless you almost know it already”.

17. Define Backward state-space search

It searches backward from the goal situation to the initial situation.

18. Differentiate between Partial Order Plan & Total order plan.

Partial-order plan

- consists partially ordered set of actions
- sequence constraints exist on these actions
- plan generation algorithm can be applied to transform partial-order plan to total-order plan

Total-order plan

- consists totally ordered set of actions

#### 19. Define action monitoring

The process of checking the preconditions of each action as it is executed, rather than checking the preconditions of the entire remaining plan. This is called action monitoring.

#### 20. Differentiate between Forward state-space search and Backward state-space search.

1. Forward state-space search : It searches forward from the initial situation to the goal situation.
2. Backward state-space search: It searches backward from the goal situation to the initial situation.

#### 21. What are the steps of planning problems using state space research methodology?

- The initial state of the search is the initial state from the planning problem. In general, each state will be a set of positive ground literals; literals not appearing are false.
- The actions that are applicable to a state are all those whose preconditions are satisfied.

The successor state resulting from an action is generated by adding the positive effect literals and deleting the negative effect literals. (In the first-order case, we must apply the unifier from the preconditions to the effect literals.) Note that a single successor function works for all planning problems—a consequence of using an explicit action representation.

- The goal test checks whether the state satisfies the goal of the planning problem.
- The step cost of each action is typically 1. Although it would be easy to allow different Costs for different actions, this is seldom done by STRIPS planners.

### **PART B**

- 1) What is Planning?
- 2) What is STRIPS?
- 3) STRIPS planning
- 4) What is Machine Learning?

### **UNIT – V EXPERT SYSTEMS**

#### **PART – A**

##### 1. Define planning.

Planning can be viewed as a type of problem solving in which the agent uses beliefs about actions and their consequences to search for a solution.

##### 2. Define Expert Systems.

A piece of software which uses databases of expert knowledge to offer advice or make decisions in such areas as medical diagnosis

##### 3. What are characteristics of Expert systems?

High performance

Understandable

Reliable

Highly responsive

4. What are the roles of Expert Systems?

Limitations of the technology

Difficult knowledge acquisition

ES are difficult to maintain

High development costs

2. Mention the roles of expert systems

In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning about knowledge, represented mainly as if-then rules rather than through conventional procedural code.

3. What is Knowledge Acquisition?

Knowledge acquisition is the process of extracting, structuring and organizing knowledge from one source, usually human experts, so it can be used in software such as an ES. This is often the major obstacle in building an ES.

4. What is Meta Knowledge?

Metaknowledge is knowledge about knowledge. . . . The hallmark of metaknowledge is its being a matter of knowing propositions in which one K[knowledge]-operator occurs within the scope of another. Knowledge about someone's ignorance thus counts as metaknowledge

5. Define Heuristics.

In computer science, artificial intelligence, and mathematical optimization, a heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

6. What is Gaze heuristics?

The gaze heuristic is a heuristic used in directing correct motion to achieve a goal using one main variable

7. What are expert shell systems?

Shells – A shell is nothing but an expert system without knowledge base. A shell provides the developers with knowledge acquisition, inference engine, user interface, and explanation facility.

8. Define MYCIN.

MYCIN was an early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient's body weight

9. Define DART.

The Dynamic Analysis and Re planning Tool, commonly abbreviated to DART, is an artificial intelligence program used by the U.S. military to optimize and schedule the transportation of supplies or personnel and solve other logistical problems

10. Define XOON.

AA is the word for 'human being'; the local name of the language is Taa ʘaan, from ʘaan 'language'. !Xoon is an ethnonym used at opposite ends of the Taa-speaking area, but not by Taa speakers in between. Most living Taa speakers are ethnic !Xoon

11. What is Knowledge?

The data is collection of facts. The information is organized as data and facts about the task domain. Data, information, and past experience combined together are termed as knowledge.

12. What is factual knowledge?

It is the information widely accepted by the Knowledge Engineers and scholars in the task domain.

13. What is Heuristic Knowledge?

It is about practice, accurate judgment, one's ability of evaluation, and guessing.

14. Mention the recommendation of user Interface.

Applies rules repeatedly to the facts, which are obtained from earlier rule application.

Adds new knowledge into the knowledge base if required.

Resolves rules conflict when multiple rules are applicable to a particular case.

15. What are the Requirements of Efficient ES User Interface?

It should help users to accomplish their goals in shortest possible way.

It should be designed to work for user's existing or desired work practices.

Its technology should be adaptable to user's requirements; not the other way round.

It should make efficient use of user input.

16. List the benefits of Expert systems.

Availability

Speed

Less production cost

Reducing risk

Steady response

17. What are all the various uses of a belief network?

- Making decisions based on probabilities in the network and on the agent's utilities.
- Deciding which additional evidence variables should be observed in order to gain useful information.

## **PART B**

1. Explain Expert System:

2. Expert system Architecture/Rule based system architecture:

3. Explain Knowledge Based System

4. Expert Systems: MYCIN

## IT6602 SOFTWARE ARCHITECTURES

### UNIT I INTRODUCTION AND ARCHITECTURAL DRIVERS

#### PART- A

##### **1. What is software architecture?**

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.”

##### **2. What is architectural pattern?**

An architectural pattern is a description of element and relation types together with a set of constraints on how they may be used.

##### **3. Define IEEE standard definition of software architecture?**

The fundamental organization of the system embodied in its components, their relationships to each other, and to the environment, and the principle guiding its design and evolution.

##### **4. Compare System Architecture and Software Architecture.**

- **Software architecture** focuses on the implementation that will solve a specific problem.
- **Systems architecture** focuses on the underlying servers (physical or virtual) and server software (web servers, database et cetera) that the software will utilize. Systems architecture is largely focused on scaling with high availability, fault tolerance and redundancy to avoid data loss.

##### **5. What is system engineering?**

System engineering means designing, implementing, deploying and operating the systems which include hardware, software and people.

##### **6. Why software architecture is important in software process?**

The software architecture gives the hierarchical structure components and their interactions. In software architecture the software model is designed and structure of that model is partitioned horizontally or vertically. This helps in implementing a quality software product.

##### **7. What is reference model?**

A reference model is a division of functionality together with data flow between the pieces. A reference model is a standard decomposition of a known problem into parts that cooperatively solve the problem.

##### **8. What is reference architecture?**

A reference architecture is a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them. Whereas a reference model divides the functionality, A reference architecture is the mapping of that functionality onto a system decomposition.

## **9. What do you mean architectural elements?**

An architectural element is a fundamental piece from which a system can be considered to be constructed. The key attributes of the architectural elements are

- a. A clearly defined set of responsibilities
- b. A clearly defined boundary
- c. A set of clearly defined interfaces, which define the services that the elements provide to the other architectural element.

## **10. Why is software architecture important?**

- Architecture is the vehicle for stakeholder communication
- Architecture manifests the earliest set of design decisions
- Architecture as a transferable, re-usable model

These are the reasons for the importance of software architecture from the technical perspective.

## **11. What are the tasks of software architect's?**

- System static partitioning and decomposition into subsystems and communications between subsystems.
- Establishing dynamic control relationships between different subsystems in data flow are controlling or message dispatching.
- Considering and evaluating all alternative architecture styles for the problem domain.

## **12. Define the term stakeholders.**

Stakeholders are the people for whom we build systems. A key part of your role as an architect knows how to work with stakeholders in order to create an architecture that meets their complex, overlapping and often conflicting needs.

## **13. What is quality attributes?**

A quality attribute is a property of a process or product that can have some qualitative or quantitative value and can be measured or observed. Quality attributes are the overall factors that affect run-time behavior, system design and user experience.

## **14. Describe the term scenario.**

A scenario is a short statement describing an interaction of one of the stakeholders with the system. An architectural scenario is a crisp, concise description of a situation that the system is likely to ace in its production environment, along with the definition of the response required of the system.

## **15. Why are qualities attributes challenging?**

- Quality attributes often impact the system as a whole.
- Quality attributes interrelate with functionality.
- Quality attributes are often in conflict with each other.

**16. List the various classes of quality attributes.**

- Qualities of the system.
- Business qualities.
- Qualities of the architecture itself.

**17. Define the term perspectives.**

Perspectives contain proven architectural knowledge and help structure the architecture process by separating concerns but focusing on cross structural quality properties rather than architectural structures.

**18. List out run time qualities.**

- Availability
- Interoperability
- Manageability
- Reliability
- Scalability
- Security

**19. Define user quality attributes.**

Usability defines how well the application meets the requirements of the user and consumer, easy to localize and globalize, providing good access for disabled users, and resulting in a good overall user experience.

**20. What is Architecture Business Cycle?**

Software architecture is a result of technical, business, and social influences. Its existence in turn affects the technical, business, and social environments that subsequently influence future architectures. We call this cycle of influences, from the environment to the architecture and back to the environment, the Architecture Business Cycle (ABC).

[OR]

It is a description of a system, used to represent relationship among structures/ components of the system to the environment in which the system is developed and implemented.

**21. List the activities in ABC.**

- Creating the business case for the system.
- Understanding the requirements.
- Creating or selecting the architecture.
- Documenting and communicating the architecture.
- Analyzing or evaluating the architecture.
- Implementing the system based on the architecture.
- Ensuring that the implementation confirms to the architecture.

**22. List out the requirements for Architecture Business Cycle.**

- Case studies
- Methods
- Techniques

**23. What are the three classes of influence that come from the developing organization?**

Three classes of influence that come from the developing organization:

- Immediate business
- Long-term business
- Organizational structure.

**24. What activities are involved in creating software architecture?**

- Creating the business case for the system
- Understanding the requirements
- Creating or selecting the architecture
- Documenting and communicating the architecture
- Analyzing or evaluating the architecture
- Implementing the system based on the architecture
- Ensuring that the implementation conforms to the architecture

**25. What is a variety of techniques for understanding requirements from the stakeholders?**

There are a variety of techniques for eliciting requirements from the stakeholders:

- Object-oriented analysis uses scenarios, or "use cases" to represent requirements.
- Safety-critical systems use more rigorous approaches, such as finite- - state-machine models or formal specification languages.
- Collection of quality attribute scenarios that support the capture of quality requirements for a system.
- Another technique that helps us understand requirements is the creation of prototypes.

**26. What is architectural structures and its types?**

A structure is the set of elements itself, as they exist in software or hardware.

The types of architectural structures are

- Module structures
- Component-and-connector structures
- Allocation structures

**27. What are Quality attribute scenarios?**

Quality attribute scenarios are the means by which quality moves from the eye of the beholder to a more objective basis.

There are 6 parts.

1. Source of Stimulus
2. Stimulus
3. Environment
4. Artifact
5. Response
6. Response Measure

**28. Define functional requirements and list its 2 characteristics. (MAY/JUNE 2016)**

- A functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.
- Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.
- A functional requirement describes *what* a software system should do.

Some of the characteristics of functional requirements are:

- Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces

**29. Write various technical constraints and business constraints. (MAY/JUNE 2016)**

**Technical Constraints in Software Architecture**

Technical constraints are fixed technical design decisions that absolutely cannot be changed.

- Programming language
- Operating system or platforms supported
- Use of a specific library or framework

**Business Constraints**

Business constraints are unchangeable business decisions that in some way restrict the software architecture design.

- Schedule
- Budget
- Team composition and make-up
- Software licensing restrictions or requirements

**30. Define functional requirements.**

Functional requirements may be calculations, Technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

**31. State nonfunctional requirements.**

Non functional requirements is the requirement that specific criteria that can be used to judge the operation of the system rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

**32. List the components of module based structure.**

- Decomposition
- Uses

- Layered
- Class or generalization

### 33. State the term architectural description?

An architectural description is a set of products that documents architecture in a way its stakeholders can understand and demonstrates that the architecture has met their concerns

### 34. Differentiate Software Architecture and Software design

Software Architecture	Software design
Define guidelines	Communicate with developers
Fundamental properties	Detailed properties
Cross cutting concerns	Details
High impact	Individual Components
Communicate with business stakeholders	Use guidelines
Manage uncertainty	Avoid uncertainty
Conceptual Integrity	Completeness

### 35. How Architectural Structures have been divided?

- Module Structures
- Component and connector Structures
- Allocation Structures

### 36. Define the term connector.

Software connectors are specialized software components that facilitate the interaction among “application” software components.

Connectors often simple procedure calls or shared data accesses, but much more sophisticated connectors are used in practice.

### 37. What is functionality?

Functionality is the ability of the system to do the work for which it was intended. For a given end user task, it would generally need different functions to cooperate and coordinate in a way as to solve the problem effectively.

## PART B

1. Explain in detail about architectural structures.
2. Describe the influences of architecture in various fields.
3. With a neat diagram explain about how ABC works and list the various activities of architecture business cycle and explain them in detail.

4. Differentiate functional and non functional requirements.
5. Explain in detail about quality attribute in various categories.
6. Discuss about influence of software architecture on organization.
7. What makes a good architectural process?
8. Explain the technical constraints and business constraints.

## **UNIT II QUALITY ATTRIBUTE WORKSHOP**

### **PART- A**

#### **1. What is software quality?**

Software **quality** is the degree to which software possesses a desired combination of attributes

#### **2. What are the properties a critical system must satisfy?**

Critical systems must satisfy security, safety, dependability, performance, and other, similar requirements as well.

#### **3. What is functionality?**

It is the ability of the system to do the work for which it was intended.

#### **4. What are the six part scenarios of a quality attribute workshop?**

QAW elicits and records six-part scenarios, where the parts include the stimulus of the scenario, the source of the stimulus, the response, the response measure, the artifact stimulated, and the environment.

#### **5. What are the inputs to QAW?**

Inputs include the

- System's business/mission drivers
- System's architectural plan

#### **6. What are the steps of QAW?**

QAW method includes the following steps:

- Business/Mission Presentation: Architectural Plan Presentation:
- Identification of Architectural Drivers
- Scenario Brainstorming
- Scenario Consolidation
- Scenario Prioritization
- Scenario Refinement

#### **7. What are the outputs of QAW? (OR) List QAW Results? (MAY/JUNE 2016)**

Outputs include a list of

- Raw scenarios
- Consolidated scenarios
- Prioritized scenarios
- Refined scenarios

## 8. What is quality attribute scenario?

A quality attribute scenario is a quality-attribute-specific requirement. It consists of six parts.

- Source of stimulus.
- Stimulus.
- Environment.
- Artifact
- Response.
- Response measure

## 9. What is security scenario?

Security is a measure of the system's ability to resist unauthorized usage while still providing its services to legitimate users. An attempt to breach security is an attack; it could be to gain access to data or services or to deny services to others

## 10. What is software testability?

Software testability refers to the ease with which software can be made to demonstrate its faults through testing. In particular, testability refers to the probability, assuming that the software has at least one fault that it will fail on its *next* test execution.

## 11. What is usability scenario?

Usability is concerned with how easy it is for the user to accomplish a desired task and the kind of user support the system provides.

## 12. What is modifiability?

Modifiability is about the cost of change. It brings up two concerns.

- What can change (the artifact)?
- When is the change made and who makes it (the environment)?

## 13. List out various business and architectural qualities. (MAY/JUNE 2016)

### BUSINESS QUALITIES

- Time to market.
- Cost and benefit.
- Projected lifetime of the system.
- Targeted market.
- Rollout schedule.
- Integration with legacy systems.

### ARCHITECTURE QUALITIES

- Conceptual integrity
- Correctness and completeness
- Buildability

## 14. What is meant by functionality and architecture? (MAY/JUNE 2016)

**Functionality:** It is the ability of the system to do the work for which it was intended.

**Software architecture** constrains its allocation to structure when *other* quality attributes are important.

**15. What is a quality attribute workshop?**

Quality attribute workshops provide a method for identifying a system's architecture critical quality attributes, such as availability, performance, security, interoperability and modifiability that are derived from mission or business goals.

**16. Explain the concept of ATAM.**

ATAM is a method for detecting risks of a complex software intensive system. It also provides insight to how design decisions affect QAs and their trade-offs.

**17. What is performance scenario?**

Performance is an indication of the responsiveness of the system to execute any action within a given time interval. It can be measured in terms of latency or throughput.

**18. Explain any two benefits of QAW.**

- Clarified quality attribute requirements
- Improved architecture documentation
- Identified risks early in the life-cycle
- Increased communication among stakeholders

**19. What are the three categories of quality assurance?**

- System qualities
- Business qualities
- Architectural qualities

**20. What is architectural scenario?**

An architectural scenario is a crisp, concise of a situation that the system is likely to face in its production environment, along with the definition of the response required of the system.

**21. What do you mean by interoperability?**

It is the quality of a system that enables it to work with other systems. It includes the quality to work with other systems not yet known.

**22. Define availability and reliability.**

Availability is an attribute that measure the proportion of the time the system is up and running.

Reliability is an attribute that measure the system's ability to continue operating over time.

**23. List the attributes related to portability.**

- Adaptability

- Installability
- Conformance
- Repalceability

**24. Mention the reasons why the use of quality attributes is not common.**

- Misunderstanding of their importance
- Inadequate languages for expressing them.
- Inadequate specification of quality requirements in projects.
- Inadequate modeling methods and notations.
- Inherent difficulty in designing for quality attributes
- Lack of documented design and architectural patterns.

**25. Mention the steps involved in QAW.**

- Introduction and QAW presentation.
- Business/ Programmatic presentation.
- Architecture plan presentation.
- Identification of architectural drivers.
- Scenario brainstorming.
- Scenario refinement.

**26. What are the six parts in a quality attribute scenario?**

1. Source- an entity that generates a stimulus
2. Stimulus- a condition that affects the system
3. Artifact- the part of the system that was stimulated by the stimulus
4. Environment- The condition under which the stimulus occurred
5. Response- The activity that results because of the stimulus.
6. Response measure- The measure by which the system's response will be evaluated.

**27. How usability scenario can be divided?**

- Learning system features
- Using a system efficiently
- Minimizing the impact of errors
- Adapting the system to user needs
- Increasing confidence and satisfaction

**28. Differentiate fault and failure.**

A fault occurs when the system no longer delivers a service that is consistent with its specification and a fault has the potential to cause a failure.

**29. Define the term tactics.**

The tactics used by the architect to create a design patterns, architectural patterns or architectural strategies. A tactic is a design decision that influences the control of a quality attribute response.

**30. What is the use of Scenario brainstorming?**

After the architectural drivers have been identified, the facilitators initiate the brainstorming process in which stakeholders generate scenarios. The facilitators review the parts of a good scenario and ensure that each scenario is well formed during the workshop.

### **31. How will you achieve desired quality attributes in the design construction?**

- Architectural styles
- Patterns
- Transforming QA into functionality
- Tactics
- Perspectives

### **32. Mention the need of documenting quality attributes.**

The QAW provides an opportunity to gather stakeholders together to provide input about their and expectations with respect to key quality attributes that are of particular concern to them.

### **33. What is the need to go for quality attribute scenarios?**

To solve solutions for problems like:

- The definitions provided for an attribute are not operational.
- A focus of discussion is often on which quality a particular aspect belongs to.
- Each attribute community has developed its own vocabulary.

### **34. What is system architecture?**

System architecture: the fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors

### **35. What is software architecture?**

The structure or structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them

### **36. What is quality workshop?**

Quality attribute workshop provides a method for identifying a systems architecture critical quality attributes, such as availability, performance, security, interoperability, that are derived from mission or business goals

### **37. Describe term Scenario?**

A scenario is a short statement describing an interaction of one of the stakeholders with the system. An architectural scenario is a crisp, concise description of a situation that the system is likely to face in its production environment, along with the definition of the system.

### **38. What are the features of concern scenario?**

A collection of concrete scenarios can be use as the quality attribute aspect to the system in requirement of the system.

a. Each scenario is concrete enough to be meaningful to the architect.

b. The details of the responses are meaningful enough so that it is possible to test whether the system has achieved the response.

### **39. What is use of scenario brainstorming?**

After the architectural drivers have been identified, the facilitators initiate the brainstorming process in which stakeholders generate scenarios. The facilitators review the parts of a good scenario and ensure a scenario is well formed during the workshop. Each stakeholder expresses a scenario representing his or her concerns with round-robin fashion.

### **40. What are the parameters of security scenario?**

- a. Confidentiality
- b. Integrity
- c. Non-repudiation
- d. Authentication
- e. Auditing

### **41. Define –Tactics.**

The tactics used by the architect to create a design using design patterns, architectural strategies. A tactic is a design decision that influences the control of a quality attribute responses.

## **PART B**

### **What is a quality attribute scenario? List the parts of such scenarios.**

1. Define quality attribute workshop. Write in detail about the steps involved in QAW.
2. Explain the roles and responsibilities of QAW.
3. How to Documenting Quality Attributes
4. Explain any one quality attribute scenario with a detailed case study.
5. Explain the various tactics available to achieve quality attributes.
6. Explain in detail about Six Part Scenario.
7. Write the standard notations we are used?
8. How to document the architecture?
9. Write short notes on Rational Unified Process/Kruchten(RUP) 4+1.
10. Explain Siemens view in detail.
11. Explain the architectural perspectives in detail.

## UNIT III ARCHITECTURAL VIEWS

### PART - A

#### 1. What is view?

A view simply represents a set of system elements and relationships among them, so whatever elements and relationships you deem useful to a segment of the stakeholder community constitute a valid view.

A view is a representation of one or more structural aspects of an architecture that illustrates how the architecture addresses one or more concerns held by one or more of its stakeholders.

#### 2. What are the categories of architectural structures?

Architectural structures are divided into three groups: module, component-and-connector (C&C), and allocation.

#### 3. What is the procedure for choosing the views for your project?

- Produce a candidate view list
- Combine views
- Prioritize

#### 4. What are the parts of a documented view?

- Primary presentation
- Element catalog
- Context diagram
- Variability guide
- Architecture background
- Glossary of terms
- Other information

#### 5. What are the main types of views?

- Logical
- Process.
- Development
- Physical

#### 6. List out the view in Siemens approach?

- Conceptual architecture view
- Module architecture view
- Execution architecture view
- Code architecture view

#### 7. What is 4+1 view of RUP?

- Logical
- Process.
- Development
- Physical
- Scenarios(use cases)

#### 8. Distinguish structures and views.

- A view is a representation of a coherent set of architectural elements, as written by and read by system stakeholders. It consists of a representation of a set of elements and the relations among them.
- A structure is the set of elements itself, as they exist in software or hardware. For example, a module structure is the set of the system's modules and their

organization. A module view is the representation of that structure, as documented by and used by some system stakeholders.

## **9. Define style.**

A *style* is a specialization of element types (e.g., “client,” “layer”) and relationship types (e.g., “is part of,” “request-reply connection,” “is allowed to use”), along with any restrictions (e.g., “clients interact with servers but not each other” or “all the software comprises layers arranged in a stack such that each layer can only use software in the next lower layer”).

## **10. List out the sections in cross view or beyond views documentation.**

Cross-view or “beyond views” documentation consists of the following sections:

- Documentation roadmap
- View template.
- System overview
- Mapping between views.
- Directory.
- Project glossary and acronym list.
- Cross-view rationale.

## **11. Define a viewpoint. (MAY/JUNE 2016)**

A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views.

## **12. How to choose views. (MAY/JUNE 2016)**

This is a three-step procedure for choosing the views for your project:

- Produce a candidate view list:
- Combine views:
- Prioritize:

## **13. Define architectural framework.**

Architectural framework not only chooses an architectural style, by identifying precisely the types of components and connectors, but also constrains the possible system configurations to particular topologies and the use of particular architectural properties.

## **14. Define the term concern.**

A concern about architecture is a requirement, an objective, an intension or an aspiration a stakeholder has for that architecture. Concerns may pertain to any aspect of the system’s functioning, development, or operation, including considerations such as performance, reliability, security, distribution.

## **15. What is an architectural element?**

An architectural element is a fundamental piece from which a system can be considered to be constructed.

## **16. What is an interface?**

An interface is a boundary across which two independent entities meet and interact or communicate with each other.

**17. Explain use of view and viewpoint.**

- View and viewpoint display related concept simultaneously.
- Both are support displaying the same data at various levels of abstraction.
- They can be adapted to the needs of specific stakeholders.

**18. What are the two critical roles played by scenario viewpoint in 4+1 view model?**

- It acts as a driver to help designers discover architectural elements during the architecture design.
- It validates and illustrates the architecture design, both on paper and as the starting point for the tests of an architectural prototype.

**19. What are the different kinds of view?**

- Module view describes how the system is to be structured as set of code units.
- Component and connector view describes how the system is to be structured as a set of interacting runtime elements.
- Allocation view describes how the system relates to non-software structure in its environment.

**20. List the choices that remain within a view type.**

- What kinds of elements are allowed?
- How they relate to each other.
- How are they used or configured.

**21. List the characteristics of the classes of logical architecture.**

- Autonomy
- Persistence
- Subordination
- Distribution

**22. What is called a view packet?**

A view packet is the smallest piece of information a stakeholder requires, which can be represented by one or more styles. For instance a single view packet could show a whole system at a high level and a number of additional view packets could be used to show the individual subsystems.

**23. Mention the benefits of using views.**

- Separation of concerns
- Communication with stakeholder groups
- Management of complexity
- Improved developer focus

**24. How will you represent views?**

- Informal notations
- Semiformal notations

- Formal notations

**25. How the 4+1 view is seen by the software industry?**

Systems engineers approach it from the physical view, then the process view. End users, customers, data specialists from the logical view. Project managers, software configuration staff see it from the development view.

**26. How an architect examines the system?**

An architect examines the system in three ways:

- Module view type.
- Component & connector view type.
- Allocation view type.

**27. How RUP helps in developing system?**

RUP is a software engineering process, which provides a disciplined approach in order to assign tasks and responsibilities in a development organization, placing particular emphasis through its 4+1 views.

**28. What do you mean by a style?**

A style is a specialization of element types (e.g., “Client” “Layer”) and relationship types (e.g., is part of, “request reply connection,” “is allowed to use”) along with any restrictions (e.g., “Clients interact with servers but not each other” or “all the software comprises layers arranged in a stack such that each layer can only use software in the next lower layer”.)

**29. Mention the styles used for logical view and development view?**

- Logical view-Object oriented style.
- Development view- Layered style.

**30. What is specific about SEI model?**

The SEI model consists of relatively independent viewpoints, compared to the other models. This allows a documentation package to be created independently for the different stakeholders. This overlaps well with the “4+1” and Siemens models.

**31. Explain use of view and view point?**

- a. View and viewpoints display related concepts simultaneously.
- b. Both are support displaying the same data at various levels of abstractions
- c. They can be adapted to the needs of specific stake holders.

**32. What are the limitations of viewpoints?**

- a. Inconsistency:
- b. Selection of the wrong set of view:
- c. Fragmentation:

## **PART B**

1. Describe 4+1 view with an example.
2. Discuss Siemens 4 views.
3. Write in detail about the various architectural perspectives.
4. Explain a view and a view point with its advantages and disadvantages.
5. Explain in detail about software structures.

## **UNIT IV ARCHITECTURAL STYLES**

### **PART- A**

#### **1. What is architectural style?**

An architectural style, sometimes called an architectural pattern, is a set of principles that provides an abstract framework for a family of systems. An architectural style improves partitioning and promotes design reuse by providing solutions to frequently recurring problems.

#### **2. What are the different types of architectural styles?**

##### **a. Data flow styles**

- Batch sequential
- Pipes and filters

##### **b. Call-and-return styles**

- Main program & subroutines
- Hierarchical layers
- OO systems

##### **c. Virtual machines styles**

- Interpreters
- Rule-based systems

##### **d. Independent components communicating processes styles**

- Event systems

##### **e. Data-centered styles (repositories)**

- Databases
- Blackboards

#### **3. What are the benefits of using architectural styles?**

- Reuse
- Understandability of system organization
- Interoperability
- Style specificity

#### **4. What is pipe and filter style?**

- Each components has set of inputs and set of outputs
- A component reads streams of data on its input and produces streams of data on its output.
- By applying local transformation to the input streams and computing incrementally, so that output begins before input is consumed. Hence, components are termed as filters.

- Connectors of this style serve as conduits for the streams transmitting outputs of one filter to inputs of another. Hence, connectors are termed pipes.

### **5. Mention the advantages of pipe and filter style.**

- The pipe-and-filter style simplifies system maintenance and enhances reuse for the same reason-filters stand alone, and we can treat them as black boxes.
- Both pipes and filters can be hierarchically composed: Any combination of filters, connected by pipes, can be packaged and appear to the external world as a filter.
- Because a filter can process its input in isolation from the rest of the system, a pipe-and-filter system is easily made parallel or distributed, providing opportunities for enhancing a system's performance without modifying it.

### **6. Mention the disadvantages of pipe and filter styles.**

- There is no way for filters to cooperatively interact to solve a problem.
- Performance in such a system is frequently poor for several reasons, all of which stem from the isolation of functionality that makes pipes and alters so modifiable; these reasons are listed below:
- Filters typically force the lowest common denominator of data representation (such as an ASCII stream). If the input stream needs to be transformed into tokens, every filter pays this parsing/unparsing overhead.
- If a filter cannot produce its output until it has received all of its input, it will require an input buffer of unlimited size. A sort filter is an example of a filter that suffers from this problem. If bounded buffers are used, the system could deadlock.

### **7. List out the goal of Call-and-Return Architectures**

Call-and-Return architectures have the goal of achieving the qualities of modifiability and solvability.

### **8. Define Remote procedure call systems. What is its goal?**

Remote procedure call systems are main-program-and-subroutine systems that are decomposed into parts that live on computers connected via a network.

The goal is to increase performance by distributing the computations and taking advantage of multiple processors.

### **9. What is layered systems?**

Layered systems are ones in which components are assigned to layers to control inter component interaction. In the pure version of this architecture, each level communicates only with its immediate neighbours.

### **10. Define event based implicit invocation.**

- Instead of invoking the procedure directly a component can announce one or more events.
- Other components in the system can register an interest in an event by associating a procedure to it.

- When the event is announced, the system itself invokes all of the procedure that have been registered for the event. Thus an event announcement “implicitly” causes the invocation of procedures in other modules.

**11. State the strengths and weakness of peer-to-peer style. (MAY/JUNE 2016)**

**Advantages**

- Easy and simple to set up only requiring a hub or a switch to connect all computers together.
- You can access any file on the computer as-long as it is set to a shared folder.
- If one computer fails to work all the other computers connected to it still continue to work.

**Disadvantages**

- Security is not good other than setting passwords for files that you don't want people to access.
- If the connections are not connected to the computers properly then there can be problems accessing certain files.
- It does not run efficient if you have many computers, it is best to used two to eight computers.

**12. Differentiate between event based style and publish-subscribe style. (MAY/JUNE 2016)**

In software architecture, publish–subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead characterize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.

Event-driven architecture (EDA), also known as message-driven architecture, is a software architecture pattern promoting the production, detection, consumption of, and reaction to events. An event can be defined as "a significant change in state".

**13. Define software connector.**

A software connector is an architectural building block tasked with effecting and regulating interactions among components.

**14. List the properties of architectural styles.**

- A vocabulary of design elements.
- They define analyses that can be performed on systems built in the style.
- A set of configuration rules.

**15. What is client server style?**

The client server architectural style describes the relationship between a client and one or more servers, where the client initiates one or more requests, waits for replies and processes the replies on receipt.

**16. What is virtual machine?**

A virtual machine implements an instruction set for an imaginary machine.

Often virtual machines are the underlying mechanism by which a programming language is executed. It specifies an interface between compiler and a real machine.

**17. State the properties of pipe and filter styles.**

- Filters do not need to know anything about what they are connected to.
- Filters can be implemented in parallel.
- Behavior of the system is the composition of behavior of the filters.
- They permit certain kinds of specialized analysis, such as throughput and deadlock analysis.

**18. List two advantages and disadvantages of layered styles.**

Advantages:

- It increases abstraction levels.
- Changes in a layer affect at most the adjacent two layers.

Disadvantages:

- Difficulty in structuring some systems in a layered fashion.
- Not universally applicable.

**19. List the key aspects of routines.**

- Routines correspond to units of the task to be performed.
- Combined through control structures.
- Routines known through interfaces.

**20. What do you mean by open and closed layered architecture?**

Open architecture:

- A layer can use services from any lower layer.
- More compact code, as the services of lower layers can be accessed directly.
- Breaks the encapsulation of layers so increase dependencies between layers.

Closed architecture:

- Each layer only uses services of the layer immediately.
- Minimizes dependencies between layers and reduces the impact of a change.

**21. What are the uses of rule based systems?**

Rule based systems are used as a way to store and manipulate knowledge to interpret information in a useful way.

Rule based systems can also be used to perform lexical analysis to compile or interpret computer programs.

**22. What are the components in blackboard system architecture?**

- The software specialist modules which are called knowledge sources.
- The blackboard a shared repository of problems partial solutions, suggestions and contributed information.
- The control shell which controls the flow of problem solving activity in the system.

**23. What are the properties of a model-view controller?**

- One central model, many views
- Each view has an associated controller
- The controller handles updates from the user of the view.

- Changes to the model are propagated to all the views.

**24. Mention some of the message delivery issues in a pub/ sub system.**

The broker in a pub/ sub system may be designed to deliver messages for a specified time, but then stop attempting delivery, whether or not it has received confirmation of successful receipt of the message by all subscribers.

**25. Explain call-return styles.**

In call-return styles, the components interact by requesting services of other components. Each component in this style provides a set of services through one or more interfaces and uses zero or more services provided by other components.

**26. List the functions of repository components.**

The repository components of a shared data system carry out a number of functions, including providing shared access to data, supporting data persistence, managing concurrent access to data, providing fault tolerance, supporting access control and handling the distribution and caching of data values.

**27. State the component of repository style.**

**Components:**

- A central data structure representing the current state of the system.
- A collection of independent components that operate on the central data structure.

**28. State the properties of Pipe and filter styles.**

**Properties:**

- Filters do not need to know anything about what they are connected to.
- Filters can be implemented in parallel.
- Behavior of the system is the composition of behavior of the filters.
- They permit certain kinds of specialized analysis, such as throughput and deadlock analysis.

**29. List the two advantages and disadvantages of layered styles.**

**Advantages:**

- It increases abstraction levels.
- Changes in a layer affect at most the adjacent two layers.

**Disadvantages:**

- Difficulty in structuring some systems in a layered fashion.
- Not university applicable.

**30. Define an architectural pattern.**

An architectural pattern is a concept that solves and delineates some essential cohesive elements of software architecture.

**31. Define software connector.**

A software connector is an architectural building block tasked with effecting and regulating interactions among components.

**32. List basic properties of architectural styles.**

- a. A vocabulary of design elements.

b. They define analyses that can be performed on systems built in that style.

c. A set of configuration rule.

**33. List functions of repository components.**

The repository components of a shared-data system carry out a number of functions, including providing shared access to data, supporting data persistence, managing concurrent access to data, providing fault tolerance, supporting access control, and handling the distribution and caching of data values.

**34. State component of repository style.**

a. A central data structure representing the current state of the system.

b. A collection of independent components that operate on the central data structure.

**35. State properties of pipes and filter styles.**

**Properties:**

a. Filters do not need to know anything about what they are connected to.

b. Filter can be implemented in parallel.

c. Behavior of the system is the composition of behavior of the filters.

d. They permit certain kinds of specialized analysis, such as throughput and deadlock analysis.

**36. List two advantages and disadvantages of layered styles.**

**Advantage:**

a. It increases abstraction levels.

b. Changes in a layer affect at most the adjacent two layers.

**Disadvantage:**

a. Difficult in structuring some system in a layered fashion.

b. Not universally applicable.

**PART B**

1. Explain the various data flow styles with suitable examples
2. Discuss the various call return styles considering suitable example.
3. Explain the various event based styles in detail.
4. Explain in detail about shared information styles.
5. Case study

## UNIT V DOCUMENTING THE ARCHITECTURE

### PART-A

#### **1. What is cloud computing?**

Cloud computing, also known as on-demand computing, is a kind of internet-based computing, where shared resources and information are provided to computers and other devices on-demand.

#### **2. What are the characteristics of cloud computing?**

- Agile
- Cost
- Device and location independence
  
- Maintenance
- Performance
- Productivity
- Reliability
- Scalability
- Security

#### **3. What are the different service models in cloud computing?**

- Infrastructure as a service
- Platform as a service
- Software as a service

#### **4. What is cloud engineering?**

Cloud engineering is the application of engineering disciplines to cloud computing. It brings a systematic approach to the high-level concerns of commercialization, standardization, and governance in conceiving, developing, operating and maintaining cloud computing systems.

#### **5. What is the purpose of Cloud Provider Interface?**

A Cloud Provider Interface (CPI) provides an abstraction from an underlying IaaS by defining a set of functions for managing virtual machines life-cycle in which might run an elastic service.

#### **6. What is service oriented architecture?**

A service-oriented architecture (SOA) is an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network.

#### **7. What are the horizontal layers of service oriented architecture?**

- Consumer Interface Layer – These are GUI for end users or apps accessing apps/service interfaces.
- Business Process Layer – These are choreographed services representing business use-cases in terms of applications.
- Services – Services are consolidated together for whole-enterprise in-service inventory.
- Service Components – The components used to build the services, such as functional and technical libraries, technological interfaces etc.

- Operational Systems – This layer contains the data models, enterprise data repository, technological platforms etc.

### 8. What are the vertical layers of service oriented architecture?

- **Integration Layer** – starts with platform integration (protocols support), data integration, service integration, application integration, leading to enterprise application integration supporting B2B and B2C.
- **Quality of Service** – Security, availability, performance etc. constitute the quality of service parameters which are configured based on required SLAs, OLAs.
- **Informational** – provide business information.
- **Governance** – IT strategy is governed to each horizontal layer to achieve required operating and capability model.

### 9. What is the good practice for documenting architecture?

Document the relevant views, and then add information that applies to more than one view, thus tying the views together.

### 10. What is an Architecture Description Language?

Modeling notation to support architecture-based development used to define and model system architecture prior to detailed design and implementation

### 11. Define ACME.

**ACME** is a simple, generic software architecture description language (ADL) that can be used as a common interchange format for architecture design tools and/or as a foundation for developing new architectural design and analysis tools.

### 12. Give the advantages and disadvantages of ADL. (MAY/JUNE 2016)

ADLs have several advantages as well as disadvantages. One **advantage** is that they are designed to represent architectures in a formal way. Another **advantage** is that they often are designed to be readable to both human and machines.

A **disadvantage** is that there is not yet an agreement of what the ADLs shall represent, especially when it comes to the behavior of the system.

### 13. How do ADL differ from other languages? (MAY/JUNE 2016)

**Architecture Description Languages (ADLs)** differ from programming languages because programming languages bind all architectural abstractions to a specific point where as ADL suppress such binding.

**Architecture Description Languages (ADLs)** differ from modeling languages because modeling Languages are more concerned with the behavior of the whole system rather than parts whereas ADLs Concentrate on the representation of the components

### 14. What is UML?

The unified modeling language (UML) is a standardized visual language for modeling software designs. UML is used different kinds of information found in software architecture documentation.

**15. What is use of sequence diagram?**

Use to show the explicit sequence of messages between architecture elements and participants of a specific trace. It can show conditional segments of the trace, loops and partial segments.

**16. What is ADL?**

An architecture description language (ADL) is used to specify the structure of a system separately from its algorithmic aspects.

**17. Define view template.**

A view template is a standard organization for a view. The purpose of a view template is that of any standard organization. It helps a reader navigate quickly to a section of interest and it helps a writer organize the information and establish criteria for knowing how much work is left to do.

**18. List the properties that ADL should exhibit.**

- Ability to represent components along with property assertions, interfaces and implementations.
- Ability to represent connectors, along with property assertions, interfaces and implementations.
- Type checking
- Ability to accommodate analysis tools

**19. What are the requirements of ADL?**

- Component abstraction
- Communication abstraction
- Communication integrity
- Ability to model dynamic architecture.
- Ability to reason about causality and time.

**20. Mention some of the benefits and drawbacks of ADL**

**Benefits:**

- Provide flexibility
- Provide high level of abstraction
- Provide testing and verification ability.
- Provide simplicity
- Provide software quality
- Reduce the cost and time.

**Drawbacks:**

- Limited operability
- Lack of automatic extensibility
- Limits the design freedom.
- Problems with concurrency control.
- Poor resource utilization.

**21. List the rules for sound documentation.**

- Documentation should be written from the point of view of the reader
- Avoid repetition
- Use a standard organization
- Record rationale
- Keep it current
- Review documentation for fitness of purpose.

**22. Mention some of the advantages of using UML.**

- UML breaks the complex system into discrete pieces the can be understood easily.
- Handover the system to new team becomes easier.
- Complex system can be understood by the disparate developers who are working on different platforms.

**23. What are the three components in visual notation?**

Hardware component represented by square corner rectangle, passive software represented by round corner rectangle, component and active software component represented by rectangle with corners cut off.

**24. What do you mean by AADL?**

Architecture analysis & Design language (AADL) is a large and complete language intended for design both the hardware and software of a system. It supports processors, buses, devices and ports as well as processes, threads and data.

**25. What do you mean by TASM?**

Timed abstract state machine (TASM) is strictly speaking not an ADL. However, it can be used to model a fairly complex system. As a name implies, it is based on the extension of timed features. A model of a system is constituted by a set of monitored variables.

**26. What are different perspectives of a system defined by UML?**

- Design
- Implementation
- Process
- Deployment

**27. What are the three kinds of interface description do a package allow?**

- The sum of all element interfaces

- A subset of element interfaces
- A separate layer interface

**28. Mention the UML building blocks.**

- Things
  - Structural-class, interface, collaboration, use case, component, node
  - Behavioral- interaction, state machine
  - Grouping- Package
- Relationships-dependency, association, generalization, realization
- Diagrams-class diagram, object diagram, use case diagram, sequence diagram, activity diagram, state chart diagram, deployment diagram, component diagram.

**29. Why document software architecture is important?**

Documenting software architecture is important because it server as the blueprint for a system and the project that develops that system. It defines work assignments and is the primary carrier of quality attributes, the best artifact for early analysis, and the key to post-deployment maintenance and mining.

**30. What is use of object diagram?**

Object diagram describe the static structure of a system at a particular time. They can be used to test class diagram for accuracy.

**31. What is role of manager in architecture document?**

Architecture documentation helps manager to form development teams based on work assignments identified, allocate project resources, and to track progress on the project.

**32. What are cloud deployment models?**

Cloud deployment models are a. Public cloud b. Private cloud c. community cloud d. Hybrid cloud

**33. List essential characteristic of cloud computing?**

On-demand self-service, ubiquitous network access, location-independent resource pooling, rapid elasticity, pay per use.

**34. What are disadvantage of UML?**

- a. Needs customization through profiles to reduce ambiguity.
- b. Difficult to assess consistency among views.
- c. Difficult to capture foreign concepts or views.

**35. What are the advantage and disadvantage of AADL.**

**Advantage:**

- a. Allows detailed specification of both hardware and software aspects of a system.
- b. Automated analysis tools check interesting end-to-end properties of system

**Disadvantage:**

- a. It requires large amount of details to capture even simple systems.
- b. Emerging tool support and UML profile support.

**PART B**

1. Explain the good practices in documenting software architecture.
2. Describe documenting views using UML.
3. What are called visual languages? Mention its advantages and disadvantages.
4. Explain about ACME.
5. Explain architectural documentation template.
6. Explain architectural description language.
7. Write a case study about service oriented architecture
8. Write a case study about cloud computing.
9. Explain Documenting Interfaces.
10. Write short notes on Documentation across Views
11. How the documentation is organized to serve a stakeholder?

# GE6757 Total Quality Management

## UNIT-I INTRODUCTION

### PART A

#### 1. Define quality.

- Quality is defined as the predictable degree of uniformity and dependability, at low cost suited to the market. (Deming).
- Quality is defined as fitness for use (Juran).
- Quality is defined as conformance to requirements (Crosby).
- Quality is totality of the characteristics of entity that bear on its ability to satisfy stated and implied needs (ISO).

#### 2. How the quality can be quantified?

$$\text{Quality} = \frac{\text{Performance}}{\text{Expectations}}$$

#### 3. Define Total Quality.

TQM is an enhancement to the traditional way of doing business. It is the art of managing the whole to achieve excellence. It is defined both a philosophy and a set of guiding principles that represent the foundation of a continuously improving organization. It is the application of quantitative methods and human resources to improve all the processes within an organization and exceed customer needs now and in the future. It integrates fundamental management techniques, existing improvement efforts, and technical tools under a disciplined approach.

#### 4. Give the Basic Concepts of TQM.

- A committed and involved management to provide long-term top-to bottom organizational support.
- An unwavering focuses on the customer, both internally and externally.
- Effective involvement and utilization of the entire work force.
- Continuous improvement of the business and production process.
- Treating suppliers as partners.
- Establish performance measures for the processes.

#### 5. List the dimensions of quality.

The dimensions of quality are

- Performance
- Futures
- Conformance
- Reliability
- Durability
- Service
- Response

- Aesthetics and
- Repetition.

## **6. What are the three components of the Juran Trilogy?**

The three components of the Juran Trilogy are

- Planning
- Control
- Improvement

## **7. What are the six basic concepts that a successful TQM programme requires?**

The six basic concepts that a successful TQM programme requires

- Top management commitment
- Focus on the customer
- Effective employee involvement
- Continuous improvement
- Treating suppliers as partners and
- Establishing performance measures.

## **8. What are the pillars of TQM?**

The four pillars of TQM are:

- Problem solving discipline
- Interpersonal skills
- Teamwork and
- Quality improvement process.

## **9. Give the Objectives of TQM.**

- To develop a conceptual understanding of the basic principles and methods associated with TQM;
- To develop an understanding of how these principles and methods have been put into effect in a variety of organizations;
- To develop an understanding of the relationship between TQM principles and the theories and models studied in traditional management;
- To do the right things, right the first time, every time.

## **10. Give the Quality Hierarchy.**

- Inspection
- Quality Control (QC)
- Quality Assurance (QA)
- Total Quality Management
- Inspect products.
- Detection.
- Finding & Fixing Mistakes.

## **11. List the tangible and intangible benefits of TQM.**

Tangible Benefits

- Improved product quality
- Improved productivity
- Reduced quality costs
- Increased market and Customers
- Increased profitability
- Reduced employee grievances  
Intangible Benefits
- Improved employee participation
- Improved teamwork
- Improved working relationships
- Improved customer satisfaction
- Improved communication
- Enhancement of job interest
- Enhanced problem-solving capacity
- Better company image.

**12. What does a typical meeting agenda contain after establishing the TQM?**

- Progress report on teams
- Customer satisfaction report
- Progress on meeting goals
- New project teams
- Recognition dinner
- Benchmarking report.

**13. Define Quality Costs.**

Quality Costs are defined as those costs associated with the non achievement of product or service quality as defined by the requirements established by the organization and its contracts with customers and society.

**14. Give the primary categories of Quality cost.**

- Preventive cost category
- Appraisal cost category
- Internal failure cost category
- External failure cost category.

**15. State the Quality Improvement Strategy.**

Reduce failure costs by problem solving Invest in the “right” prevention activities Reduce appraisal costs where appropriate and in a statistically sound manner continuously evaluate and redirect the prevention effort to gain further quality improvement.

**16. Define Quality Planning.**

A quality plan sets out the desired product qualities and how these are assessed and define the most significant quality attributes. It should define the quality assessment process. It should set out which organizational standards should be applied and, if necessary, define new standards.

**17. What is needed for a leader to be effective?**

To be effective, a leader needs to know and understand the following:

- People, paradoxically, need security and independence at the same time.
- People are sensitive to external rewards and punishments and yet are also strongly self-motivated.
- People like to hear a kind word of praise.
- People can process only a few facts at a time; thus, a leader needs to keep things simple.
- People trust their gut reaction more than statistical data.
- People distrust a leader's rhetoric if the words are inconsistent with the leader's actions.

### **18. What is the important role of senior management?**

Listening to internal and external customers and suppliers through visits, focus groups and surveys communication. To drive fear out of the organization, break down barriers, remove system roadblocks, anticipate and minimize resistance to change and in general, change the culture.

### **19. What is needed for a leader to be effective?**

To be effective, a leader needs to know and understand the following:

- People, paradoxically, need security and independence at the same time.
- People are sensitive to external rewards and punishments and yet are also strongly self-motivated.
- People like to hear a kind word of praise.
- People can process only a few facts at a time; thus, a leader needs to keep things simple.
- People trust their gut reaction more than statistical data.
- People distrust leader's rhetoric if the words are inconsistent with the leader's actions.

### **20. Give the basic steps to strategic quality planning.**

- Customer needs
- Customer positioning
- Predict the future
- Gap analysis
- Closing the gap
- Alignment
- Implementation

## **PART B**

- 1) What is quality cost? Explain the techniques used for Quality cost.
- 2) Explain the principles of TQM.
- 3) Discuss the 14 principles of Deming Philosophy.
- 4) Elaborate the barriers to TQM implementation.
- 5) Elucidate the concepts of Leadership.
- 6) Enlighten the basic concepts of TQM.
- 7) Write the benefits of TQM.

## UNIT-II TQM PRINCIPLES

### PART A

#### **1. Who are internal and external customers?**

The customers inside the company are called internal customers, whereas the customers outside the company are called external customers.

#### **2. What are the customer's perceptions on quality?**

The six important customer's perceptions are:

- Performance
- Features
- Service
- Warranty
- Price and
- Reputation.

#### **3. List the various tools used for collecting customer complaints.**

The various tools used are:

- Comment card.
- Customer questionnaire
- Focus groups
- Toll-free telephone numbers
- Report cards
- The Internet and computer etc.

#### **4. What is meant by customer retention?**

Customer retention is the process of retaining the existing customers.

#### **5. What is motivation?**

Motivation means a process of stimulation people to accomplish desired goals.

#### **6. What are the Maslow's basic needs?**

Maslow's basic needs are:

- Physiological
- Safety
- Society
- Esteem and
- Self-actualization needs.

#### **7. What are physiological needs?**

Physiological needs are the biological needs required to preserve human life. These needs include needs for food, clothing and shelter.

#### **8. List the Herzberg's motivators and dissatisfies.**

Motivator factors Dissatisfied or hygiene factors

- Achievement
- Recognition
- The work itself
- Responsibility
- Advancement and growth.
- Supervisors
- Working conditions
- Interpersonal relationships
- Pay and security
- Company policy and
- Administration

### **9. Define empowerment.**

Empowerment is an environment in which people have the ability, the confidence, and the commitment to take the responsibility and ownership to improve the process and initiate the necessary steps to satisfy customer's requirements within well defined boundaries in order to achieve organizational values and goals.

### **10. What are the conditions necessary for empowerment?**

The conditions required are:

1. Everyone must understand the need for change.
2. The system needs to change to the new paradigm.
3. The organization must provide information, education and skill to its employees.

### **11. Define team and teamwork.**

- A team can be defined as a group of people working together to achieve common objectives or goals.
- Teamwork is the cumulative actions of the team during which each member of the team subordinates his individual interests and opinions to fulfill the objectives or goals of the group.

### **12. List the different types of teams.**

The different types of teams are

- Process improvement team.
- Cross-functional team.
- Natural work team.
- Self-directed work team.

### **13. Name different members in a team.**

The different members in a team are

1. Team leader

2. Facilitator
3. Recorder
4. Timekeeper and
5. Member

**14. What is needed for a leader to be effective?**

To be effective, a leader needs to know and understand the following:

- People, paradoxically, need security and independence at the same time.
- People are sensitive to external rewards and punishments and yet are also strongly self-motivated.
- People like to hear a kind word of praise.
- People can process only a few facts at a time; thus, a leader needs to keep things simple.
- People trust their gut reaction more than statistical data.
- People distrust a leader's rhetoric if the words are inconsistent with the leaders actions.

**15. What is the important role of senior management?**

- Listening to internal and external customers and suppliers through visits, focus groups and surveys.
- Communication.
- To drive fear out of the organization, break down barriers, remove system roadblocks, anticipate and minimize resistance to change and in general, change the culture.

**16. Give the basic steps to strategic quality planning.**

- Customer needs
- Customer positioning
- Predict the future
- Gap analysis
- Closing the gap
- Alignment
- Implementation

**17. What is meant by recognition in an organization?**

Recognition is a process whereby management shows acknowledgement of a employee's outstanding performance.

**18. Classify rewards.**

**1. Intrinsic rewards:** These are related to feelings of accomplishment or self worth.

**2. Extrinsic rewards:** These are related to pay or compensation issues.

**19. What is performance appraisal?**

Quality Costs are defined as those costs associated with the non achievement of product or service quality as defined by the requirements established by the organization and its contracts with customers and society.

**20. List the tools used for feedback.**

- Comment cards
- Surveys
- Focus groups
- Toll-free telephone lines
- Customer visits
- Report cards
- The internet
- Employee feedback
- American Customer Satisfaction Index

**21. Give the steps involved in training process.**

The steps involved in training process are

- Make everyone aware of what the training is all about.
- Get acceptance.
- Adapt the program.
- Adapt to what has been agreed upon.

**22. Define Recognition and Reward.**

- Recognition is a form of employee motivation in which the organization
- Publicly acknowledges the positive contributions an individual or team has made to the success of the organization.
- Reward is something tangible to promote desirable behavior. Recognition and reward go together to form a system for letting people know they are valuable
- Members of the organization.

**23. What are the types of appraisal formats?**

The types of appraisal formats are

- Ranking
- Narrative
- Graphic
- Forced choice

**24. What are the benefits of employee involvement?**

**The benefits of employee involvement are**

- Employee Involvement improves quality and increases productivity because
- Employees make better decisions using their expert knowledge of the Process.
- Employees are more likely to implement and support decisions they had a part in making.
- Employees are better able to spot and pinpoint areas for improvement.

**25. What are the basic ways for a continuous process improvement?**

The basic ways for a continuous process improvement are

- Reduce resources
- Reduce errors
- Meet or exceed expectations of downstream customers
- Make the process safer

- Make the process more satisfying to the person doing it.

**26. What are the three components of the Juran Trilogy?**

The three components of the Juran Trilogy are

- Planning
- Control
- Improvement

**27. What are the steps in the PDSA cycle?**

The steps in the PDSA cycle are

The basic Plan-Do-Study-Act is an effective improvement technique.

- Plan carefully what is to be done
- Carry out the plan
- Study the results
- Act on the results by identifying what worked as planned and what didn't.

**28. What are the phases of a Continuous Process Improvement Cycle?**

The phases of a Continuous Process Improvement Cycle are

- Identify the opportunity
- Analyze the process
- Develop the optimal solutions
- Implement
- Study the results
- Standardize the solution
- Plan for the future

**29. What are the three key elements to a partnering relationship?**

The three key elements to a partnering relationship are

- Long-term commitment
- Trust
- Shared vision

**30. What are the objectives of Performance measures?**

The objectives of Performance measures are

- Establish baseline measures and reveal trends.
- Determine which processes need to be improved.
- Indicate process gains and losses.
- Compare goals with actual performance.
- Provide information for individual and team evaluation.
- Provide information to make informed decisions.
- Determine the overall performance of the organization.

**31 What are the characteristics used to measure the performance of a particular process?**

The characteristics used to measure the performance of a particular Process are

- Quantity

- Cost
- Time
- Accuracy
- Function
- Service
- Aesthetics

### 32. Define 5S.

5S Philosophy focuses on effective work place organization and standardized work procedures. 5S simplifies your work environment, reduces waste and non-value activity while improving quality efficiency and safety.

- **Sort** – (Seiri) the first S focuses on eliminating unnecessary items from the workplace.
- **Set In Order** (Seiton) is the second of the 5Ss and focuses on efficient and effective storage methods.
- **Shine:** (Seiso) once you have eliminated the clutter and junk that has been clogging your work areas and identified and located the necessary items, the next step are to thoroughly clean the work area.
- **Standardize:** (Seiketsu) once the first three 5S"s have been implemented, you should concentrate on standardizing best practice in your work area.
- **Sustain:** (Shitsuke) this is by far the most difficult S to implement and achieve. Once fully implemented, the 5S process can increase morale, create positive impressions on customers, and increase efficiency and organization.

### 33. What is a Kaizen?

Kaizen is a Japanese word for the philosophy that defines management's role in continuously encouraging and implementing small improvements involving everyone. It is the process of continuous improvement in small increments that make the process more efficient, effective, under control and adaptable.

## **PART B**

- 1) Explain Juran trilogy for Continuous Process Improvement.
- 2) Discuss the concept of PDSA cycle.
- 3) Elucidate the Kaizen principle.
- 4) How the employee will be involved in doing a process? Explain.
- 5) List and explain Quality statement.
- 6) Explain customer satisfaction model.
- 7) Enlighten the concept of employee involvement.

## UNIT III TQM TOOLS & TECHNIQUES I

### PART A

#### 1. Give the seven tools of quality.

- Pareto Diagram
- Process Flow Diagram
- Cause-and-Effect Diagram
- Check Sheets
- Histogram
- Control Charts
- Scatter Diagrams.

#### 2. Define Statistics.

Statistics is defined as the science that deals with the collection, tabulation, analysis, interpretation, and presentation of quantitative data.

#### 3. What is a measure of central tendency?

A measure of central tendency of a distribution is a numerical value that describes the central position of the data or how the data tend to build up in the center. There are three measures in common in use in quality via, the average, the median and the mode.

#### 4. What is Measures of dispersion?

Measures of dispersion describe how the data are spread out or scattered on each side of the central value. The measures of dispersion used are range and standard deviation.

#### 5. What is a normal curve?

The normal curve is a symmetrical, unimodal, bell-shaped distribution with the mean, median and mode having the same value.

#### 6. What is the use of the control chart?

The control chart is used to keep a continuing record of a particular quality characteristic. It is a picture of process over time.

#### 7. Give the objectives of the attribute charts.

- Determine the average quality level.
- Bring to the attention of management any changes in the average.
- Improve the product quality.
- Evaluate the quality performance of operating and management personnel.
- Determine acceptance criteria of a product before shipment to the customer.

#### 8. Define Six Sigma Problem Solving Method.

- **Define** - improvement opportunity with an emphasis on increasing customer satisfaction.
- **Measure** - determine process capability (Cp/ Cpk) & dpmo (defects per million)

opportunities).

- **Analyze** - identify the vital few process input variables that affect key product output variables (“Finding the knobs”).
- **Improve** - Make changes to process settings, redesign processes, etc. to reduce the number of defects of key output variables.
- **Control** - Implement process control plans, install real-time process monitoring tools, and standardize processes to maintain levels.

#### **9. What are the new seven management tools?**

- Affinity Diagram
- Interrelationship Digraph
- Tree Diagram
- Matrix Diagram
- Prioritization Matrices
- Process Decision Program Chart
- Activity Network diagram

#### **10. Define Benchmarking.**

Benchmarking is a systematic method by which organizations can measure themselves against the best industry practices. The essence of benchmarking is the process of borrowing ideas and adapting them to gain competitive advantage. It is a tool for continuous improvement.

#### **11. Enumerate the steps to benchmark.**

- Decide what to benchmark
- Understand current performance
- Plan
- Study others
- Learn from the data
- Use the findings.

#### **12. What are the types of benchmarking?**

- Internal
- Competitive
- Process

#### **13. What are the four basic steps included in SPC?**

The four basic steps included in SPC are

- Measuring the process
- Eliminating variances in the process to make it consistent.
- Monitoring the process.
- Improving the process to its best target value.

#### **14. Mention the seven basic tools involved in statistic quality control.**

The seven tools involved in statistical quality control. They are,

- Pareto diagram
- Check sheet
- Cause and effect diagrams
- Scatter diagram

- Histogram
- Control charts
- Graphs

**15. What is Pareto chart?**

A Pareto chart is a special form of a bar graph and is used to display the relative importance of problems or conditions.

**16. Give some applications of Pareto chart.**

The applications of Pareto chart are,

- Focusing on critical issues by ranking them in terms of importance and frequency (Example: which course causes the most difficulty for students?; which problem with product X is most significant to our customers?)
- Prioritizing problems or causes to efficiently initiate problem solving (Example: which discipline problems should be tackled first? or what is the most frequent complaint by parents, regarding the school? solution of what production problem will improve quality most?)

**17. What is the use of SPC?**

SPC is used to monitor the consistency of processes used to manufacture a product as designed.

**18. Define check sheet. Mention its uses.**

The check sheet is a data gathering and interpretation tool. A check sheet is used for,

- Distinguishing between fact and opinion (Example: How does the community perceive the effectiveness of the school in preparing students for the world of work?)
- Gathering data about how often a problem is occurring? (Example: How often are students missing classes?)
- Gathering data about the type of problem occurring. (Example: What is the most common type of word processing error created by the students-grammar, punctuation, transposing letter etc.?)

**19. What are the uses of cause and effect diagram?**

A cause and effect diagram is used for,

- Identifying potential causes of a problem or issue in an orderly way. (Example: why membership in the band decreased? Why isn't the phone4 being answered on time? Why is the production process suddenly producing so many defects?)
- Summarizing major causes under four categories. (Example: People, machines, methods and materials or policies, procedures, people and plant.)

**20. What is scatter diagram?**

A scatter diagram is used to interpret data by graphically displaying the relationship between two variables.

**21. List some applications of scatter diagram.**

The applications of scatter diagram

- Validating hunches about a cause-and-effect relationship between types of variables (examples: I wonder if students who spend more time watching TV having higher or lower average GPA's? IS there a relationship between the production speed of an n operator and the number of defective parts made? Is there relationship between typing speed in WPM and errors made?)
- Displaying the direction of the relationship (positive negative, etc). (Examples: will test scores increase or decrease if the students spend more time in study hall? Will increasing assembly line speed, increase or decrease the number of defective parts made? Do faster typists make more or fewer typing errors?)
- Defective parts produced? How strong is the relationship between typing faster and the number of typing errors made?).

## **22. Define histogram.**

A histogram is used to display in bar graph format measurement data distributed by categories.

## **23. What are the problems that can be interpreted by the histogram?**

The problems that can be interpreted by the histogram are,

- Skew problems
- Clustering problems.

## **24. Define control chart.**

Control chart is defined as a display of data in the order that they occur with statistically determined upper and lower limits of expected common cause variations. It is used to indicate special causes of process variations to monitor a process for maintenance and to determine if process changes have has the desired effect.

## **25. What is line graph?**

A line graph is a way to summaries how two pieces of information are related and how they vary depending on one another. The numbers along a side of the line graph are called the scale.

## **26. What is an arrow diagram?**

An arrow diagram is another term for a PERT or CPM chart. It is graphic descriptions of the sequential steps that must be completed before a project can complete.

## **27. Give some applications of arrow diagram.**

The applications of arrow diagram are,

- Understanding and managing complex project or task.
- Understanding and managing a project that is of major importance to the organization, and the consequences of late completion are sever.
- Understanding and managing a project in which multiple activities must take place and be managed simultaneously.
- Explaining the project status to others.

## **28. How is an arrow diagram constructed?**

Steps in constructing an arrow diagram are,

- Select a team that is knowledgeable about the project, its task and subtasks.
- Record all of the tasks and subtasks necessary to the completion of the project.
- Sequence the tasks.
- Assign time duration to each task.
- Calculate the shortest possible implementation time schedule using the critical path method.
- Calculate the earliest starting and finishing times for each task.
- Locate tasks with slack (extra) time and calculate total slack.
- Update the schedule as the project is being completed.

### **29. What is nominal group technique?**

The nominal group technique is a structured process, which identifies and ranks the major problems or issues that need addressing.

## **PART B**

- 1) Elaborate the Seven traditional tools of quality.
- 2) Explain the Seven Management Tools.
- 3) Enlighten the concepts of Six Sigma.
- 4) Discuss the Benchmarking concept and its phases.
- 5) List and explain the types of Benchmarking.
- 6) Elucidate the FMEA.
- 7) Explain the stages of FMEA.

## **UNIT IV TQM TOOLS & TECHNIQUES II**

### **1. What is a QFD?**

Quality Function Deployment is a planning tool used to fulfill customer expectations. It is a disciplined approach to product design, engineering, and production and provides in-depth evaluation of a product.

### **2. What are the benefits of QFD?**

- Customer driven
- Reduces implementation time
- Promotes teamwork
- Provides documentation.

### **3. What are the steps required to construct an affinity diagram?**

- Phrase the objective
- Record all responses
- Group the responses
- Organize groups in an affinity diagram.

### **4. What are the goals of TPM?**

The overall goals of Total Productive Maintenance, which is an extension of TQM are

- Maintaining and improving equipment capacity

- Maintaining equipment for life
- Using support from all areas of the operation
- Encouraging input from all employees
- Using teams for continuous improvement.

**5. Give the seven basic steps to get an organization started toward TPM.**

- Management learns the new philosophy
- Management promotes the new philosophy
- Training is funded and developed for everyone in the organization
- Areas of needed improvement are identified
- Performance goals are formulated
- An implementation plan is developed
- Autonomous work groups are established.

**6. What are the major loss areas?**

- Planned downtime
- Unplanned downtime
- Idling and minor stoppages
- Slow-downs
- Process nonconformities
- Scrap

**7. Define TPM.**

T: Total = All encompassing by maintenance and production individuals working together.

P: Productive = Production of goods and services that meet or exceed customer's expectations.

M: Maintenance = Keeping equipment and plant in as good as or better than the original condition at all times.

**8. Define quality cost.**

Quality cost is defined as the cost associated with the non-achievement of product/service quality as defined by the requirements established by the organization and its contracts with customers and society.

**9. List the categories of quality costs.**

The categories of quality cost are

- Cost of prevention
- Cost of appraisal
- Cost of internal failures and
- Cost of external failures.

**10. What is meant by cost of prevention?**

Prevention costs are the costs that are incurred on preventing a quality problem from arising.

### **11. List the elements of cost of prevention.**

The elements of cost of prevention are

- Cost of quality planning
- Cost of documenting
- Process control cost
- Cost of training
- Costs associated with preventing recurring defects.

### **12. What is cost appraisal?**

Appraisal costs are the costs that are incurred in assessing that the products/services conform to the requirements.

### **13. What are the costs of appraisal?**

The costs of appraisal are

- Cost of receiving test and equipment
- Cost of Laboratory acceptance testing
- Cost of installation testing
- Cost of installation and commissioning
- Cost of maintenance and calibration of testing and inspecting equipments.

### **14. What is meant by cost of internal failures?**

The costs associated with defective products, components and materials that fail to meet quality requirements and result in manufacturing losses are called as costs of internal failures. These costs are linked to correcting mistakes before delivery of the product

### **15. List the components cost of internal failures.**

The costs of internal failures are

- Cost associate with scrap and rejects.
- Cost of repair and rework.
- Cost of design changes.
- Cost of trouble shooting
- Cost of re-inspection and retesting etc.

### **16. What is meant by cost of external failures?**

It consists of the costs which are generated because of defective products being shipped to customers. These costs are associated with the adjustments of malfunctions after delivery of the product.

### **17. Give the sub-elements of Preventive cost category.**

- Marketing/Customer/User
- Product/Service/Design development
- Purchasing
- Operations
- Quality Administration
- Other Prevention Costs.

**18. Give the sub-elements of Appraisal cost category.**

- Purchasing appraisal cost
- Operations appraisal cost
- External appraisal cost
- Review of test and application data
- Miscellaneous quality evaluations.

**19. Give the sub-elements of internal failure cost category.**

- Product or Service Design costs (Internal)
- Purchasing failure costs
- Operations failure costs.

**20. Give the sub-elements of External failure cost category.**

- Complaint investigations of customer or user service
- Returned goods
- Retrofit and recall costs
- Warranty claims
- Liability costs
- Penalties
- Customer or user goodwill
- Lost sales
- Other external failure costs.

**21. Give the typical cost bases.**

- Labor
- Production
- Unit
- Sales.

**22. How will you determine the optimum cost?**

- Make comparison with other organizations
- Optimize the individual categories
- Analyze the relationships among the cost categories.

**PART B**

- 1) Explain the Bench marking Process and reasons to Benchmark.
- 2) Discuss the QFD process.
- 3) Elaborate the House of Quality in Quality Function Deployment.
- 4) Elucidate the Quality circles.
- 5) Explain the organizational structure.
- 6) Discuss the concept of TPM.
- 7) How the performance can be measured?

# UNIT V QUALITY SYSTEMS

## PART A

### 1. Give the ISO 9000 Series of Standards.

- ISO 9000, “Quality Management and Quality Assurance Standards Guidelines for Selection and Use”.
- ISO 9001, “Quality Systems – Model for Quality Assurance in Design, Development, Production, Installation & Servicing”.
- ISO 9002, “Quality Systems – “Model for Quality Assurance in Production, Installation & Servicing”.
- ISO 9003, “Quality Systems – “Model for Quality Assurance in Final Inspection and Test”.
- ISO 9004-1, “Quality Management and Quality System Elements – Guidelines”.

### 2. What is the need for ISO 9000?

ISO 9000 is needed to unify the quality terms and definitions used by industrialized nations and use terms to demonstrate a supplier’s capability of controlling its processes.

### 3. Give some other quality systems.

**The quality systems are**

- QS-9000
- QS-9000
- TE-9000
- AS9000

### 4. Enumerate the steps necessary to implement the Quality Management System.

The steps necessary to implement the Quality Management System are

- Senior management commitment
- Appoint the management representative
- Awareness
- Appoint an implementation team
- Training
- Time schedule
- Select element owners
- Review the present system
- Write the documents
- Install the new system
- Internal audit
- Management review
- Pre assessment
- Registration.

### 5. What are the three sections of QS-9000?

The three sections of QS-9000 are

- Common requirements, which include the exact text of ISO 9001 and the addition of automotive/heavy trucking requirements.
- Additional requirements covering production part approval process, continuous improvement and manufacturing capabilities.
- Customer-specific requirements.

**6. Give the objectives of the internal audit.**

The objectives of the internal audit

- Determine the actual performance conforms to the documented quality systems.
- Initiate corrective action activities in response to deficiencies.
- Follow up on noncompliance items of previous audits.
- Provide continued improvement in the system through feedback to management.
- Cause the audit to think about the process, thereby creating possible improvements.

**7. What are the requirements of ISO 14001?**

The requirements of ISO 14001 are

- General requirements
- Environmental policy
- Planning
- Implementation and operation
- Checking and corrective action
- Management review.

**8. What are the benefits of ISO 14000?**

The benefits of ISO 14000 are

- Global
- Facilitate trade and remove trade barriers
- Improve environmental performance of planet earth
- Build consensus that there is a need for environment management and a common terminology for EMS.
- Organizational.

**9. What are the four elements for the checking & corrective action of ISO 14001?**

- Monitoring and measuring
- Nonconformance and corrective and preventative action
- Records
- EMS audit.

**10. What are the seven elements for the implementation & operations of ISO 14001?**

- Structure and responsibility
- Training, awareness and competency
- Communication
- EMS documentation
- Documentation control
- Operational control
- Emergency preparedness and response

**11. What are the four elements for the planning of ISO 14001?**

- Environmental aspects
- Legal and other requirements
- Objectives and targets
- Environmental Management Programs.

**12. Give the types of Organizational Evaluation Standards.**

- Environmental Management System
- Environmental Auditing
- Environmental Performance Evaluation.

**13. Give the types of Product Evaluation Standards.**

- Environmental Aspects in Product Standards
- Environmental Labeling
- Life-Cycle Assessment.

**14. Define Quality Audits.**

Quality Audits examine the elements of a quality management system in order to evaluate how well these elements comply with quality system requirements.

**15. Give the usage of an effective recognition and reward system.**

- Serves as a continual reminder that the organization regards quality and productivity as important.
- Offers the organization a visible technique to thank high achievers for outstanding performance.
- Provides employees a specific goal to work toward. It motivates them to improve the process.
- Boosts morale in the work environment by creating a healthy sense of competition among individuals and teams seeking recognition.

**16. What are the typical measurements frequently asked by managers and teams?**

- Human Resource
- Customers
- Production
- Research & Development
- Suppliers
- Marketing/Sales
- Administration.

**17. List the ISO/QS 9000 elements.**

- Management responsibility
- The Quality system
- Contract review
- Design control
- Document and data control
- Purchasing
- Control of customer-supplied product
- Product identification and traceability
- Process control

- Inspection and testing
- Control of inspection, measuring and test equipment
- Inspection and test status.

**18. What are the benefits of ISO?**

- Fewer on-site audit by customers.
- Increased market share.
- Improved quality, both internally and externally.
- Improve product and service quality levels from suppliers.
- Greater awareness of quality by employees.
- A documented formal system.
- Reduced operating costs.

**19. Give the ISO 9001 requirements.**

- Scope
- Normative Reference
- Terms and Definitions
- Quality Management System
- Management Responsibility
- Resource Management
- Product Realization

**20. What are the methods of actual audit?**

- Examination of documents.
- Observation of activities.
- Interviews.

**PART B**

- 1) List and explain the elements of ISO 9000:2000.
- 2) Explain the implementation and documentation of Quality System.
- 3) Elaborate the requirements of ISO 14000.
- 4) Enlighten the Benefits of ISO 14000.
- 5) Discuss about ISO9000:2000 Quality Systems.
- 6) Why is ISO 9000 important?
- 7) Discuss the ISO 8204 for quality.
- 8) Explain key processes of Software Quality Management.