ORACLE

-TAMIL TELEGRAM TECH INTERVIEWERS.

Q #1) What is Oracle and what are its different editions?

Answer: Oracle is one of the popular databases provided by Oracle Corporation, which works on relational management concepts and hence it is referred to as Oracle RDBMS as well. It is widely used for online transaction processing, data warehousing, and enterprise grid computing.

Q #2) How will you identify Oracle Database Software Release?

Answer: Oracle follows a number of formats for every release.

For Example,

Release 10.1.0.1.1 can be referred to as:

10: Major DB Release Number
1: DB Maintenance Release Number
0: Application Server Release Number
1: Component Specific Release Number
1: Platform Specific Release Number

Q #3) How will you differentiate between VARCHAR & VARCHAR2?

Answer: Both VARCHAR & VARCHAR2 are Oracle data types that are used to store character strings of variable length. Their differences are:

- VARCHAR can store characters up to 2000 bytes while VARCHAR2 can store up to 4000 bytes.
- VARCHAR will hold the space for characters defined during declaration even if all of them are not used whereas VARCHAR2 will release the unused space.

Q #4) What is the difference between TRUNCATE & DELETE command?

Answer: Both the commands are used to remove data from the database.

The difference between the two include:

- TRUNCATE is a DDL operation while DELETE is a DML operation.
- TRUNCATE removes all the rows but leaves the table structure intact. It can not be rolled back as it issues COMMIT before and after the command execution while the DELETE command can be rolled back.
- The TRUNCATE command will free the object storage space while the DELETE command does not.

- TRUNCATE is faster compared to DELETE.

Q #5) What is meant by RAW datatype?

Answer: RAW datatype is used to store variable-length binary data or byte strings.

The difference between RAW & VARCHAR2 datatype is that PL/SQL does not recognize this data type and hence, cannot do any conversions when RAW data is transferred to different systems. This data type can only be queried or inserted in a table.

Syntax: RAW (precision)

Q #6) What is meant by Joins? List the types of Joins.

Answer: Joins are used to extract data from multiple tables using some common columns or conditions.

There are various types of Joins as listed below:

- INNER JOIN
- OUTER JOIN
- CROSS JOINS or CARTESIAN PRODUCT
- EQUI JOIN
- ANTI JOIN
- SEMI JOIN

Q #7) What is the difference between SUBSTR & INSTR functions?

Answer:

- SUBSTR function returns the sub-part identified by numeric values from the provided string.
  - **For Example,** [SELECT SUBSTR ('India is my country', 1, 4) from dual] will return "Indi".
- INSTR will return the position number of the sub-string within the string.
  - **For Example,** [SELECT INSTR ('India is my country', 'a') from dual] will return 5.

Q #8) How can we find out the duplicate values in an Oracle table?

Answer: We can use the below example query to fetch the duplicate records.

```
SELECT EMP_NAME, COUNT (EMP_NAME)
FROM EMP
GROUP BY EMP_NAME
HAVING COUNT (EMP_NAME) &gt; 1;
```

Q #9) How does the ON-DELETE-CASCADE statement work?

Answer: Using ON DELETE CASCADE will automatically delete a record in the child table when the same is deleted from the parent table. This statement can be used with Foreign Keys.

We can add ON DELETE CASCADE option on an existing table using the below set of commands.

Syntax:

```
ALTER TABLE CHILD_T1 ADD CONSTRAINT CHILD_PARENT_FK REFERENCES
PARENT_T1 (COLUMN1) ON DELETE CASCADE;
```

Q #10) What is an NVL function? How can it be used?

Answer: NVL is a function that helps the user to substitute value if null is encountered for an expression.

It can be used as the below syntax.

```
NVL (Value_In, Replace_With)
```

Q #11) What is the difference between a Primary Key & a Unique Key?

Answer: Primary Key is used to identify each table row uniquely, while a Unique Key prevents duplicate values in a table column.

Given below are a few differences:

- The primary key can be only one on the table while unique keys can be multiple.
- The primary key cannot hold null value at all while the unique key allows multiple null values.
- The primary key is a clustered index while a unique key is a non-clustered index.

Q #12) How TRANSLATE command is different from REPLACE?

Answer: TRANSLATE command translates characters one by one in the provided string with the substitution character. REPLACE command will replace a character or a set of characters with a complete substitution string.

For Example:

```
TRANSLATE ('Missisippi','is','15') =&gt; M155151pp1
REPLACE ('Missisippi','is','15') =&gt;  M15s15ippi
```

Q #13) How can we find out the current date and time in Oracle?

Answer: We can find the current date & time using SYSDATE command in Oracle.

Syntax:

```
SELECT SYSDATE into CURRENT_DATE from dual;
```

Q #14) Why do we use COALESCE function in Oracle?

Answer: COALESCE function is used to return the first non-null expression from the list of arguments provided in the expression. There must be a minimum of two arguments in an expression.

Syntax:

```
COALESCE (expr 1, expr 2, expr 3…expr n)
```

Q #15) How will you write a query to get 5th RANK students from the table STUDENT_REPORT?

Answer: The query will be as follows:

```
SELECT TOP 1 RANK
FROM (SELECT TOP 5 RANK
FROM STUDENT_REPORT
```

```
ORDER BY RANK DESC) AS STUDENT
ORDER BY RANK ASC;
```

Q #16) When do we use the GROUP BY clause in SQL Query?

Answer: GROUP BY clause is used to identify and group the data by one or more columns in the query results. This clause is often used with aggregate functions like COUNT, MAX, MIN, SUM, AVG, etc.

Syntax:

```
SELECT COLUMN_1, COLUMN_2
FROM TABLENAME
WHERE [condition]
GROUP BY COLUMN_1, COLUMN_2
```

Q #17) What is the quickest way to fetch the data from a table?

Answer: The quickest way to fetch the data would be to use ROWID in the SQL query.

Q #18) Where do we use DECODE and CASE Statements?

Answer: Both DECODE & CASE statements will function like IF-THEN-ELSE statement and they are the alternatives for each other. These functions are used in Oracle to transform the data values.

For Example:

DECODE Function

```
Select ORDERNUM,
DECODE (STATUS,'O', 'ORDERED','P', 'PACKED,'S','SHIPPED','A','ARRIVED')
FROM ORDERS;
```

CASE Function

```
Select ORDERNUM
, CASE (WHEN STATUS ='O' then 'ORDERED'
WHEN STATUS ='P' then PACKED
WHEN STATUS ='S' then 'SHIPPED'
ELSE 'ARRIVED') END
FROM ORDERS;
```

Both the commands will display order numbers with their respective status as,

If,

Status O= Ordered
Status P= Packed
Status S= Shipped
Status A= Arrived

Q #19) Why do we need integrity constraints in a database?

Answer: Integrity constraints are required to enforce business rules so as to maintain the integrity of the database and prevent the entry of invalid data into the tables. With the help of the below-mentioned constraints, relationships can be maintained between the tables.

Various integrity constraints are available which include Primary Key, Foreign Key, UNIQUE KEY, NOT NULL & CHECK.

Q #20) What do you mean by MERGE in Oracle and how can we merge two tables?

Answer: The MERGE statement is used to merge the data from two tables. It selects the data from the source table and inserts/updates it in the other table based on the condition provided in the MERGE query.

Syntax:

```
MERGE INTO TARGET_TABLE_1
USING SOURCE_TABLE_1
ON SEARCH_CONDITION
WHEN MATCHED THEN
INSERT (COL_1, COL_2…)
VALUES (VAL_1, VAL_2…)
WHERE &lt;CONDITION&gt;
WHEN NOT MATCHED THEN
UPDATE SET COL_1=VAL_1, COL_2=VAL_2…
WHEN &lt;CONDITION&gt;
```

Q #21) What is the use of Aggregate functions in Oracle?

Answer: Aggregate functions perform summary operations on a set of values to provide a single value. There are several aggregate functions that we use in our code to perform calculations. **These are:**

- AVG
- MIN
- MAX
- COUNT
- SUM
- STDEV

Q #22) What are the set operators UNION, UNION ALL, MINUS & INTERSECT meant to do?

Answer: The set operator facilitates the user to fetch the data from two or more than two tables at once if the columns and relative data types are the same in the source tables.

- **UNION** operator returns all the rows from both the tables except the duplicate rows.
- **UNION ALL** returns all the rows from both the tables along with the duplicate rows.
- **MINUS** returns rows from the first table, which does not exist in the second table.
- **INTERSECT** returns only the common rows in both the tables.

Q #23) Can we convert a date to char in Oracle and if so, what would be the syntax?

Answer: We can use the TO_CHAR function to do the above conversion.

Syntax:

```
SELECT to_char (to_date ('30-01-2018', 'DD-MM-YYYY'), 'YYYY-MM-DD') FROM
dual;
```

Q #24) What do you mean by a database transaction & what all TCL statements are available in Oracle?

Answer: Transaction occurs when a set of SQL statements are executed in one go. To control the execution of these statements, Oracle has introduced TCL i.e. Transaction Control Statements that use a set of statements.

The set of statements include:

- **COMMIT:** Used to make a transaction permanent.
- **ROLLBACK:** Used to roll back the state of DB to last the commit point.
- **SAVEPOINT:** Helps to specify a transaction point to which rollback can be done later.

Q #25) What do you understand by a database object? Can you list a few of them?

Answer: Object used to store the data or references of the data in a database is known as a database object. The database consists of various types of DB objects such as tables, views, indexes, constraints, stored procedures, triggers, etc.

Q #26) What is a nested table and how is it different from a normal table?

Answer: A nested table is a database collection object, which can be stored as a column in a table. While creating a normal table, an entire nested table can be referenced in a single column. Nested tables have only one column with no restriction of rows.

For Example:

```
CREATE TABLE EMP (
EMP_ID NUMBER,
EMP_NAME  TYPE_NAME)
```

Here, we are creating a normal table as EMP and referring a nested table TYPE_NAME as a column.

Q #27) Can we save images in a database and if yes, how?

Answer: BLOB stands for Binary Large Object, which is a data type that is generally used to hold images, audio & video files or some binary executables. This datatype has the capacity of holding data up to 4 GB.

Q #28) What do you understand by database schema and what does it hold?

Answer: Schema is a collection of database objects owned by a database user who can create or manipulate new objects within this schema. The schema can contain any DB objects like table, view, indexes, clusters, stored procs, functions, etc.

Q #29) What is a data dictionary and how can it be created?

Answer: Whenever a new database is created, a database-specific data dictionary gets created by the system. This dictionary is owned by the SYS user and maintains all the metadata related to the database. It has a set of read-only tables and views and it is physically stored in the SYSTEM tablespace.

Q #30) What is a View and how is it different from a table?

Answer: View is a user-defined database object that is used to store the results of an SQL query, which can be referenced later. Views do not store this data physically but as a virtual table, hence it can be referred to as a logical table.

CLOUD COMPUTING

-TAMIL TELEGRAM TECH INTERVIEWERS.

Q1). What is cloud computing?

Cloud computing is a new-age computer technology that is internet-based. It is the next-generation technology that utilizes web-based clouds to provide the services whenever and wherever the users need them.

Q2). What are the benefits of cloud computing?

The major advantages of cloud computing are:

- *Data backup*
- *Storage of data.*
- *Powerful server capabilities.*
- *Incremental productivity.*
- *Very cost-effective and time-saving.*
- *Software as Service is known as SaaS.*

Q3). What is a cloud?

A cloud is an amalgamation of hardware, network, services, storage, and interfaces that aid in delivering computing as a service.

It has three users :

- *End users*
- *Business management users*
- *cloud service provider*

Q4). Mention platforms that are used for large-scale cloud computing?

Apache Hadoop – Apache Hadoop is known to be an open-source software platform for dispersed storage and distributed dispensation of huge data sets on computer bundles built from the product hardware. Hadoop services supply data storage, data access, data processing, data governance, operations, and security.

MapReduce – Google has definitely revolutionized analysis of large-scale datasets with this great platform. It enables the processing of massive datasets using cloud sources and other commodity hardware. It provides for fault forbearance and clear sociability at the software level.

Q5). What is PaaS?

PaaS is one of the categories of cloud computing that provides a platform and environment to let the developers build apps and services over the internet. PaaS services are hosted in the cloud and are accessed by users simply through their web browsers.

Q6). What is SaaS?

Software as a service (**SaaS**) is a software distribution model in which a third-party provider hosts applications and makes them accessible to customers over the Internet.

Q7). What is IaaS?

Infrastructure as a service (**IaaS**) is a form of cloud computing that provides for virtualized computing resources over the internet itself.

Q8). What is CaaS?

Communications as a Service (**CaaS**) is an outsourced scheme interactions solution that can be leased from a single vendor over the web.

Q9). What are the basic characteristics of cloud computing?

- *There are four basic characteristics of cloud computing: Self-service provisioning and automatic de-provisioning.*
- *Elasticity and scalability.*
- *Billing self-service-based usage model. Standardized interfaces.*

Q10). What are the essential things that must be followed before going to a cloud computing platform

The essential things that must be followed are-

- *Compliance issues*
- *Data storage types*

- *Maintaining data integrity in the cloud*
- *Ensuring availability and access*
- *Protection from loss of data*
- *Business continuity*
- *Uptime – reduction of downtime*

11. What is the classification of Cloud Computing services?

Cloud computing services are classified into three broad categories namely:

- *Infrastructure As A Service (Iaas),*
- *Platform As A Service (Paas), And*
- *Software As A Service (Saas).*
- *These services are sometimes called the "cloud computing stack", for the reason that they are built on top of one another.*

Cloud Computing Interview Questions And Answers For Professionals

For the professionals who are looking to attend Cloud Computing interviews recently, here are some of the most popular interview questions and answers that will help you in the right way. Over here, we have included the top frequently asked questions with answers to help the freshers as well as the experienced professionals in the field.

Cloud Computing Interview Questions And Answers For Experienced

Q12). What are the different layers that define cloud architecture?

The different layers used by cloud architecture are

- *CLC or Cloud Controller*
- *Cluster Controller*
- *Walrus*
- *NC or Node Controller*
- *SC or Storage Controller*

Q13). Mention About Applications Now A Days?

Topmost cloud computing apps include google docs which are very fast and secure. There is also an extended mobile version of google docs app so that you can have access to your data from your smartphone. Pixlr and Phoenix, jaycut also are some of the apps used for cloud computing.

Q14). What are the types of data used in cloud computing?

There are many different data types used in cloud computing like emails, images, contracts, blogs, etc. As it is a known fact that data is increasing day by day so it is essential for the new data types to securely store these new data types. For instance, if you wish to store video then you need a new data type.

Q15). How can a company benefit from cloud computing?

A company can benefit from cloud computing in the following ways-

- *More secure data backup and data storage*
- *Software as a service (SaaS)*
- *Take advantage of powerful server capabilities without hardware investment*
- *Platform and OS agnostic*
- *Better positioning for growth and scale*
- *Increased productivity*
- *Sandboxing and virtualization capabilities*
- *Cost-effectiveness*

Q16). How can you best secure data for transportation to the cloud?

First of all make sure that your data cannot be blocked as it moves into the cloud storage by making sure that there are absolutely no data leaks —malevolent or not, from its cloud storage. This can be easily accomplished by using a secure key.

Q17). What is Security management in terms of Cloud Computing?

Here are the popular security management processes in terms of cloud computing-

- ***Identity management-*** *It provides access to the authorization of application services.*
- ***Access control permission-*** *It provides users to have complete controlling access of another user too who is entering into the same cloud environment.*
- ***Authentication and Authorization-*** *It provides access to only the authorized and authenticated personnel to securely access the data and applications.*

Q18). Mention is the difference between elasticity and scalability in cloud computing?

Scalability is a typical characteristic of cloud computing which is used to handle the escalating workload by escalating in proportion to the amount of resource capacity. By the use of scalability, the architecture provides resources on requirement BA is resources as and when the requirement is being raised by the traffic. On the other hand, Elasticity is a characteristic that provides for the concept of commissioning and decommissioning of the huge amount of resource capacity dynamically. It is usually measured by the speed by which the resources are coming on demand and the usage of those resources.

Q19). What are Hybrid Clouds?

Hybrid clouds are nothing but a combination of public clouds as well as private clouds. It is usually preferred over both clouds because it applies the most healthy approach to the implementation of the cloud architecture. It includes the functionalities as well as the features of both worlds at the same time. It allows the organizations to design their own cloud and permit them to give control over to someone else as well.

Q20). What is "EUCALYPTUS" in cloud computing?

EUCALYPTUS is an acronym that stands for Elastic Utility Computing Architecture For Linking Your Program To Useful Systems. It is used to execute clusters in cloud computing platforms.

Q21). What is the difference between cloud computing and mobile computing?

Mobile computing and cloud computing are slightly similar concepts. Mobile computing uses the concept of cloud computing. Cloud computing, on the other hand, provides its users with the data which they require while migrating or working in mobile computing for the apps running on the remote server systems and gives the user access for storage and management of data.

NETWORKING BASICS

1. What is meant by a link and node?

A network includes two or more computers connected directly by a physical medium like coaxial cable or optical fiber. Link is the physical medium of connection in this setup, and nodes are the computers connected.

2. Define IP address.

In a network system, an IP address is a unique software address of a computer. It is 32 bit.

3. What do you understand by DNS?

There are two types of server/client programs. One is directly used by the user and the other support application programs. Domain Name system belongs to the second type

as it is used by other programs, for example, to find the IP address of an e-mail recipient.

4. What is a peer-to-peer process?

A peer-to-peer process refers to all processes on a machine that communicates at a given layer.

5. Define network topology.

Network topology refers to the network's physical structure that defines how nodes or computers will be connected.

6. What is a firewall?

A firewall is a security system concept that helps in protecting computers from any cyber-attack or unauthorized access.

7. Tell us the maximum segment length of the 100Base-FX network.

The maximum length of a 100Base-FX network is 412 meters.

8. What is the role of the network layer in the OSI reference model?

The network layer is responsible for packet switching, control of network congestion, and data routing. This layer has routers operating under it.

9. Explain OSI and its role in computer networks.

OSI or Open Systems Interconnect is a reference model for data communication. It has seven layers, each defining a particular aspect of how network devices communicate and connect. One layer dictates how data is transmitted, while the other deals with physical media used.

10. Give the disadvantage of the peer-peer network.

As the resources to be accessed are shared by one of the workstations on the network, there is a performance hit.

11. Define ping.

Ping is a utility program that allows us to check connectivity on the network between network devices. A device can be pinged by using its device name(like computer name) or IP address.

12. What is meant by clustering support?

Clustering support is the ability of a network operating system in a fault-tolerant group to connect multiple servers. The primary purpose of clustering is that if one server fails, the processing can continue with the next server in the cluster.

13. How does dynamic host configuration protocol help in network administration?

The network administrator applies the dynamic host configuration protocol to create a pool of IP addresses instead of visiting each client computer to configure a static IP address. This pool is known as the scope that can be assigned to clients dynamically.

14. What do you understand by decoder?

The decoder is a type of circuit that converts the digital signal into an analog signal and encoded data into its original format.

15. Can you tell us about the use of Pseudo TTY?

It is a false terminal allowing external machines to log in or to connect through Telnet. No connection can take place without it.

16. Tell us about the advantages of a Modem.

Some advantages of modem are:

- Its speed depends on the cost
- It is more helpful in connecting LAN with the Internet
- It is the most widely used data communication roadway.

17. Explain Proxy Server and its function.

IP addresses are required for data transmission and are even used by DNS to route to the correct website. Without knowledge of the actual and correct IP address, it is not possible to identify the network's physical location. Proxy servers prevent unauthorized access of IP addresses and make the computer network virtually invisible to external users.

18. What are the characteristics of networking?

The characteristics of networking are:

- Medium- the channel used by computers for communication
- Topology- the way computers are arranged in the network physically or logically

- Protocols- deals with how computers communicate with one another.

19. What do you understand by beaconing?

When a network self-repair its issues, then it is known as beaconing. It is mainly used in Fiber Distributed Data Interface (FDDI) and token ring networks. If a device in the network faces any problem, then the devices that are not receiving any signal are notified. This way, the problem gets repaired within the network.

20. What is SLIP?

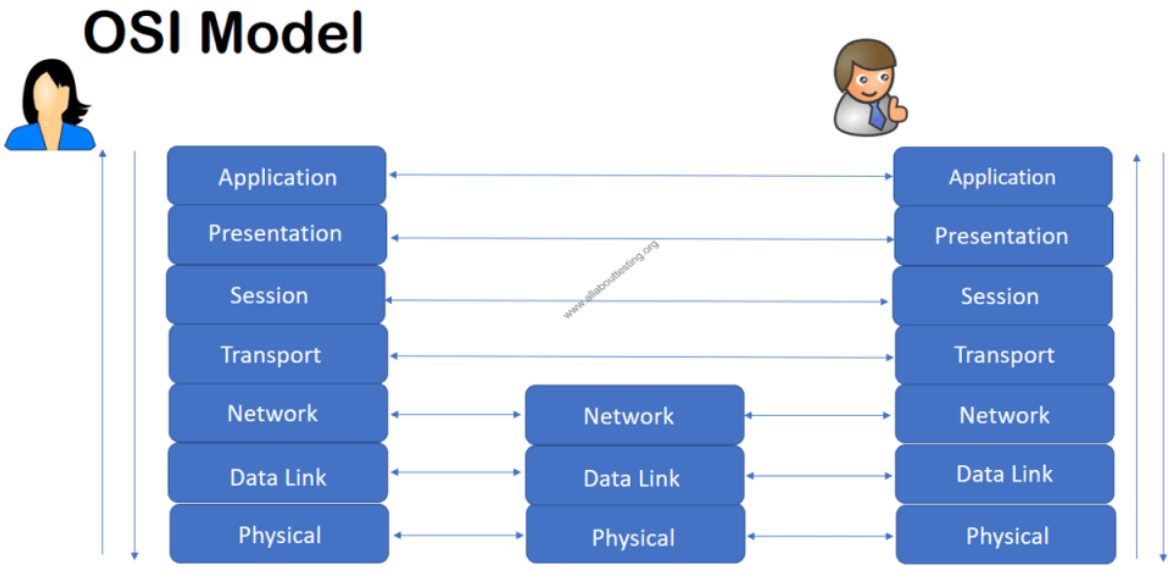
SLIP refers to Serial Line Interface Protocol. It is used for transmitting IP datagrams over a serial line.

OSI LAYER

Q1. Define OSI layers.

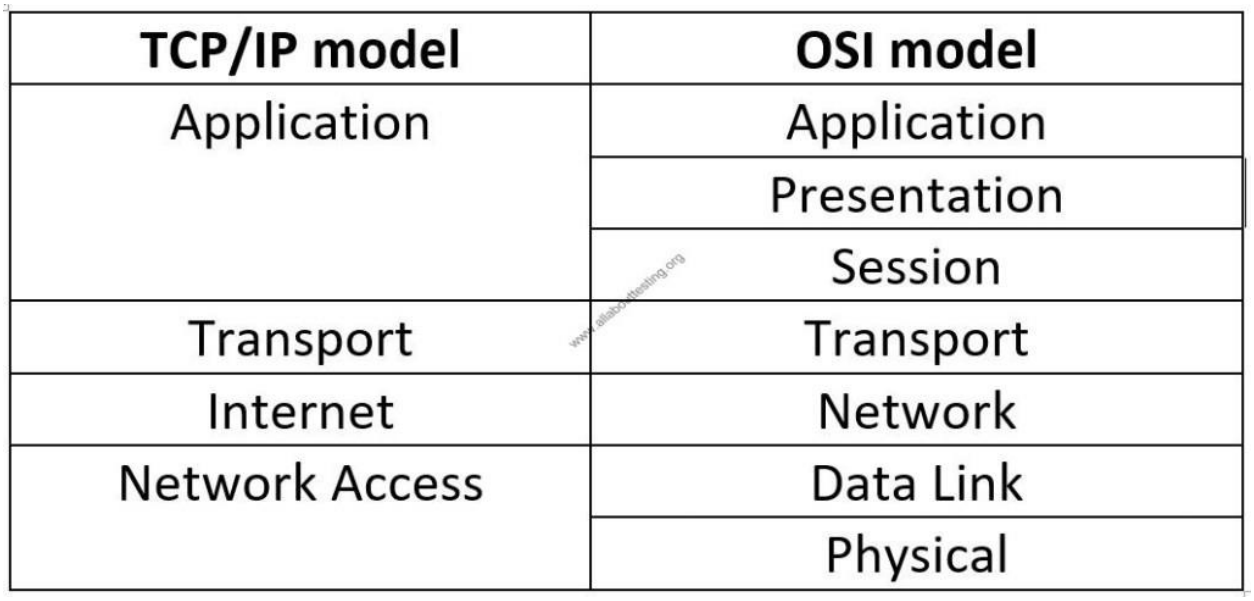
Ans: OSI stands for Open System Interconnection. There are 7 layers in the OSI model and each layer has a different capability. OSI model helps networking professionals in understanding information flow from one source to destination. Although the OSI model not performing any function in the networking process.

Remember, all devices and software applications use OSI model to explain data flow between source and destination.



Q2. Are there any alternative models to the OSI model? If yes, define it.

Ans: TCP/IP is the alternate model that also explains the information flow in the network. It is a simpler representation compared to the OSI model but contains fewer details of protocols than the OSI model.



TCP/IP model	OSI model
Application	Application
	Presentation
	Session
Transport	Transport
Internet	Network
Network Access	Data Link
	Physical

Q3. What is the difference between TCP and UDP?

Ans: TCP and UDP: Comparison between Two Transport Protocols

	TCP	UDP
Acronym	Transmission Control Protocol	User Datagram Protocol
Connection	Connection Oriented Protocol	Connection Less Protocol
Function	Message transfer from source to destination in an ordered and error-checked stream.	Message transfer from one point to another without checking any order and any error in the stream.
Usage	High reliability More transmission time	Low reliability Less transmission time
Reliability	Guarantee Data transfer and arrives in the same order in which it was sent.	No guarantee that the messages or all packets sent would reach from source to destination.
Other protocols	HTTP, HTTPs, FTP, SMTP, Telnet	DNS, DHCP, TFTP, SNMP, RIP, VOIP
Header size	TCP header size is 20 bytes	UDP header size is 8 bytes
Headers	Fields in TCP Header 1. Sequence Number, 2. AcK number, 3. Data offset, 4. Reserved, 5. Control bit, 6. Window, 7. Urgent Pointer 8. Options, 9. Padding,	Fields in UDP Header 1.Length, 2.Source port, 3.Destination port, 4.CheckSum

	10. CheckSum, 11. Source port, 12. Destination port	
Handshake	Three-way handshake	No handshake
Data Flow Control	It controls the flow of data	Does not have an option for flow control of data

Showing 1 to 10 of 10 entries

PreviousNext

Q4. What is the importance of the Physical Layer in the OSI model?

Ans: Physical layer resembles the actual transfer of information from source to destination in the form bitstream – electrical impulse, light, or radio signal. In simple words, it accepts a frame from the data link layer and converts it into bits. It also accepts bits from the physical medium and converts it into the frame.
Common protocols for this layer are EIA/TIA-232, EIA/TIA-449, X.21, HSSI, V.24, V.35, and SONET.

Q5. Which layers perform error detection and flow control?

Ans: On receiving and while transmission of information, Layer 2 – Data Link layer decoded and encoded data into bits. The data link layer is further divided into two sublayers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. This layer also performs error checking and flow control.

Q6. How the network administrator detect the problem?

Ans: Network administrators use the OSI model to understand the information flow and try to find problems by further understanding each layer's protocols. Experience in networking reduces the time to find problems and resolve them. Network problems may be a loose physical connection, configuration issues, etc.

Q7. What is the difference between flow control and error control?

Ans: As the name suggests, flow control controls the rate of information transmitted to ensure the receiver's efficient delivery of data. While error control checks and correct errors in the data bits and packets.

Q8. What is Data encapsulation?

Ans: Data encapsulation is a process of adding extra information at each layer of the OSI model while information flow from one host to another host. Information such as source and destination address, protocol information, type of data etc.

Q9. What are the differences between the MAC sublayer and LLC sublayer?

Ans: MAC sublayer stands for Media Access Control layer. MAC address works on Layer 2-Data Link Layer. This layer controls the permission of data to transmit it.

LLC sublayer stands for Logical Link Control layer. This layer controls frame synchronization, flow control, and error checking.

Q10. What is the difference between half-duplex and full-duplex?

Ans: In half-duplex, information can flow in both directions but not simultaneously. While in full-duplex, information can flow in both directions simultaneously.

OBJECT ORIENTED PROGRAMMING

- BY TAMIL TELEGRAM TECH INTERVIEWERS

Q #1) Explain in brief what do you mean by Object Oriented Programming in Java?

Answer: OOP deals with objects, like real-life entities such as pen, mobile, bank account which has state (data) and behavior (methods).

With help of access, specifiers access to this data and methods is made secured. Concepts of encapsulation and abstraction offer data hiding and access to essentials, inheritance, and polymorphism help code reuse and overloading/overriding of methods and constructors, making applications platform-independent, secured and robust using languages like Java.

Q #2) Explain Is Java a pure Object Oriented language?

Answer: Java is not an entirely pure object-oriented programming language. **The following are the reasons:**

- Java supports and uses primitive data types such as int, float, double, char, etc.
- Primitive data types are stored as variables or on the stack instead of the heap.
- In Java, static methods can access static variables without using an object, contrary to object-oriented concepts.

Q #3) Describe class and object in Java?

Answer: Class and object play an integral role in object-oriented programming languages like Java.

- Class is a prototype or a template that has state and behavior supported by an object and used in the creation of objects.
- The object is an instance of the class, **for example,** Human is a class with the state as having a vertebral system, brain, color, and height and has behavior such as canThink(), ableToSpeak(), etc.

Q #4) What are the differences between class and objects in Java?

Answer: Following are a few major differences between class and objects in Java:

Class	Object
Class is a logical entity	Object is physical entity
Class is a template from which object can be created	Object is an instance of the class
Class is a prototype that has the state and behavior of similar objects	Objects are entities that exist in real life such as mobile, mouse, or intellectual objects such as bank account
Class is declared with class key word like class Classname { }	Object is created via new keyword as Employee emp = new Employee();
During class creation, there is no allocation of memory	During object creation, memory is allocated to the object
There is only one-way class is defined using the class keyword	Object creation can be done many ways such as using new keyword, newInstance() method, clone() and factory method.
Real-life examples of Class can be a •A recipe to prepare food. •Blue prints for an automobile engine.	Real-life examples of Object can be •A food prepared from recipe. •Engine constructed as per blue-prints.

Q #5) Why is a need for Object-oriented programming?

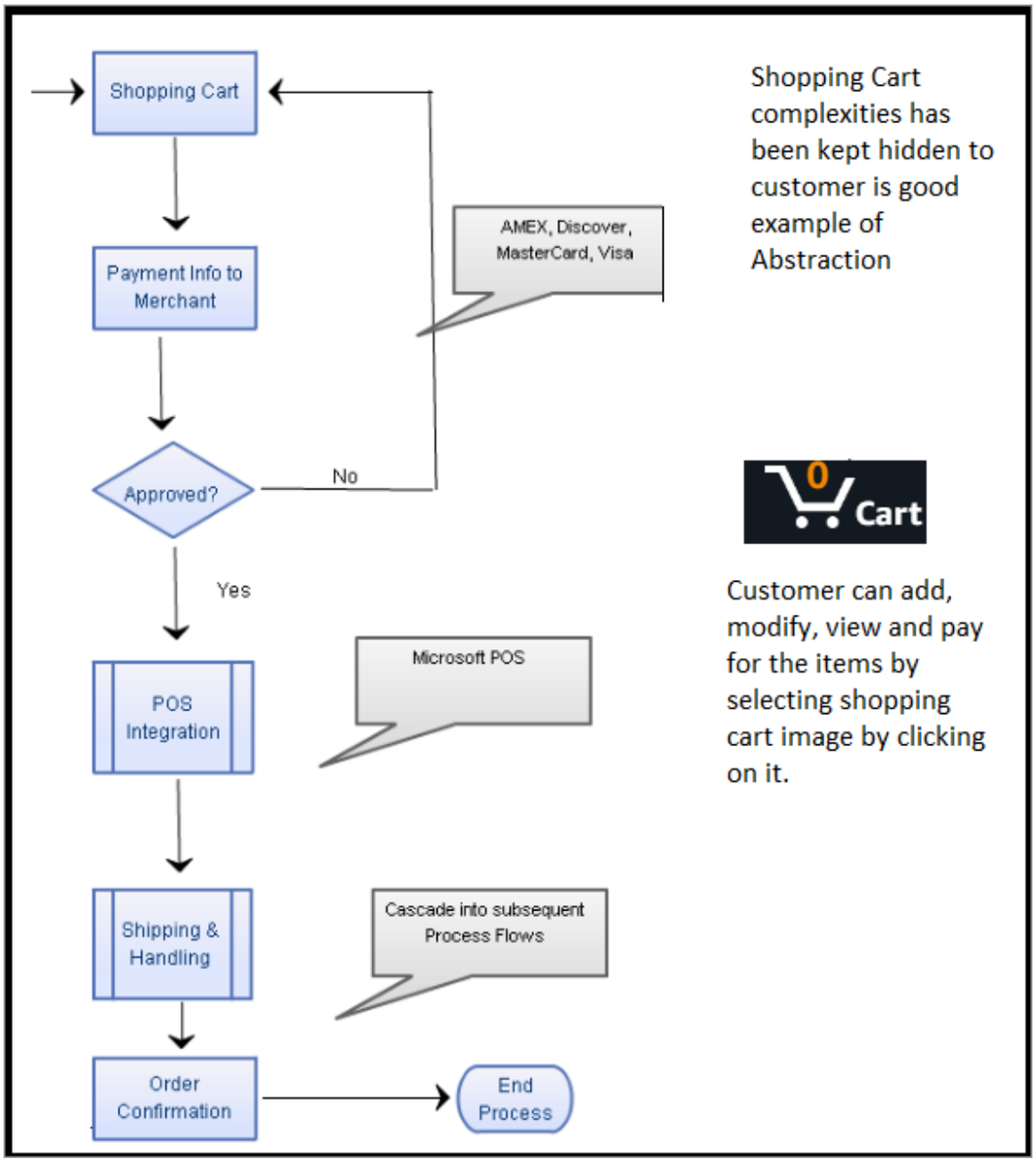
Answer: OOP provides access specifiers and data hiding features for more security and control data access, overloading can be achieved with function and operator overloading, Code Reuse is possible as already created objects in one program can be used in other programs.

Data redundancy, code maintenance, data security, and advantage of concepts such as encapsulation, abstraction, polymorphism, and inheritance in object-oriented programming provide an advantage over previously used procedural programming languages.

Q #6) Explain Abstraction with a real-time example.

Answer: Abstraction in object-oriented programming means hiding complex internals but to expose only essential characteristics and behavior with respect to context. In real life, an example of abstraction is an online shopping cart, say at any e-commerce site. Once you select a product and book order, you are just interested in receiving your product on time.

How things happen is not what you are interested in, as it is complex and kept hidden. This is known as abstraction. Similarly, take the example of ATM, the complexity of internals of how money is debited from your account is kept hidden, and you receive cash via a network. Similarly for cars, how petrol makes the engine run the automobile is extremely complex.



Q #7) Give some real-time examples and explain Inheritance.

Answer: Inheritance means one class (sub class) acquiring properties of another class (super class) by inheritance. In real life, take an example of inheritance of a normal bicycle where it is a parent class and a sports bike can be a child class, where sports bike has inherited properties and behavior of rotating wheels with pedals via gears that of a normal bike.

Q #8) How polymorphism works in Java, explain with real-life examples?

Answer: Polymorphism is an ability to have multiple forms or capability of the method to do different things. In real life, the same person performing different duties behaves differently. In-Office he is an employee, at home, he is a father, during or in after school tuitions he is a student, on weekends he plays cricket and is a player in the playground.

In Java, there are two types of polymorphism

- **Compile-time polymorphism:** This is achieved by method overloading or operator overloading.
- **Runtime polymorphism:** This is achieved by method overriding.

Q #9) How many types of inheritance are present?

Answer: Various types of inheritance are listed below:

- **Single Inheritance:** Single child class inherits characteristics of the single-parent class.
- **Multiple Inheritance:** One class inherits features of more than one base class and is not supported in Java, but the class can implement more than one interface.
- **Multilevel Inheritance:** A class can inherit from a derived class making it a base class for a new class, **for example,** a Child inherits behavior from his father, and the father has inherited characteristics from his father.
- **Hierarchical Inheritance:** One class is inherited by multiple subclasses.
- **Hybrid Inheritance:** This is a combination of single and multiple inheritances.

Q #10) What is Interface?

Answer: Interface is similar to the class where it can have methods and variables, but its methods do not have a body, just a signature known as the abstract method. Variables declared in the interface can have public, static, and final by default. Interface is used in Java for abstraction and multiple inheritances, where the class can implement multiple interfaces.

Q #11) Can you explain the advantages of Abstraction and Inheritance?

Answer: Abstraction reveals only essential details to the user and ignores or hides irrelevant or complex details. In other words, data abstraction exposes the interface and hides implementation details. Java performs abstraction with the help of interfaces and abstract classes. Advantage of abstraction is that it makes simple in viewing things by reducing or hiding the complexity of implementation.

Duplication of code is avoided, and it increases code reusability. Only essential details are revealed to the user and improves the security of the application.

Inheritance is where child class inherits functionality (behavior) of the parent class. We need not write code once written in parent class for functionality again in the child class and thus making it easier to reuse the code. The code becomes readable as well. Inheritance is used where there “is a” relation. **Example:** Hyundai **is a** car OR MS Word **is a** software.

Q #12) What is the difference between extends and implements?

Answer: Both extends and implements keyword are used for inheritance but in different ways.

The differences between Extends and Implements keywords in Java are explained below:

Extends	Implements
A class can extend another class (child extending parent by inheriting his characteristics). Interface as well inherit (using keyword extends) another interface.	A class can implement an interface
Sub class extending super class may not override all of the super class methods	Class implementing interface has to implement all the methods of the interface.
Class can only extend a single super class.	Class can implement any number of interfaces.
Interface can extend more than one interfaces.	Interface cannot implement any other interface.
Syntax: **class Child extends class Parent**	**Syntax:** **class Hybrid implements Rose**

Q #13) What are different access modifiers in Java?

Answer: Access modifiers in Java controls access scope of class, constructor, variable, method, or data member. **Various types of access modifiers are as follows:**

- **Default access modifier** is without any access specifier data members, class and methods, and are accessible within the same package.
- **Private access modifiers** are marked with the keyword private, and are accessible only within class, and not even accessible by class from the same package.
- **Protected access modifiers** can be accessible within the same package or subclasses from different packages.
- **Public access modifiers** are accessible from everywhere.

Q #14) Explain the difference between abstract class and method?

Answer: Following are some differences between abstract class and abstract method in Java:

Abstract Class	Abstract Method
Object cannot be created from the abstract class.	Abstract method has a signature but does not have a body.
Sub class created or inherit abstract class to access members of abstract class.	It is compulsory to override abstract methods of super class in their sub class.
Abstract class can contain abstract methods or non abstract methods.	Class containing abstract method should be made abstract class.

Q #15) What are the differences between method and constructor?

Answer: Following are the differences between constructors and methods in Java:

Constructors	Methods
Constructors name should match with that of Class.	Methods should not have same name as Class name.
They are used to create, initialize and allocate memory to the object.	Methods are used to execute certain statements written inside them.
Constructors are implicitly invoked by the system whenever objects are created.	Methods are invoked when it is called.
They are invoked using new keyword while creating an instance of the class (object).	Methods are invoked during program execution.
Constructor does not have return type.	Method has a return type.
Constructor cannot be inherited by the subclass.	Methods can be inherited by a sub class.

Q #16) What is a constructor in Java?

Answer: Constructor is a method without a return type and has its name the same as the class name. When we create an object, a default constructor allocates memory for an object during the compilation of Java code. Constructors are used to initializing objects and set initial values for object attributes.

Q #17) How many types of constructors can be used in Java? Please explain.

Answer: There are basically three types of constructors in Java.

These are:

1. **Default constructor:** This constructor is without any parameter and invokes every time you create an instance of a class (object). If a class is an Employee, then the syntax of the default constructor will be Employee().
2. **No-arg constructor:** As the name implies, a constructor without any argument is called a no-arg constructor.
3. **Parameterized constructor:** Constructor with a number of parameters is called a parameterized constructor. You are required to provide arguments, i.e. initial values with respect to the data type of parameters in that constructor.

Q #18) Why new keyword is used in Java?

Answer: When we create an instance of class, i.e. objects, we use the Java keyword **new**. It allocates memory in the heap area where JVM reserve space for an object. Internally, it invokes the default constructor as well.

Syntax:

```
Class_name obj = new Class_name();
```

Q #19) When do you use the super keyword?

Answer: Super is a Java keyword used to identify or refer parent (base) class.

- We can use super to access super class constructor and call methods of the super class.
- When method names are the same in super class and sub class, to refer super class, the **super** keyword is used.
- To access the same name data members of parent class when they are present in parent and child class.
- **Super** can be used to make an explicit call to no-arg and parameterized constructors of the parent class.
- Parent class method access can be done using **super**, when child class has method overridden.

Q #20) When do you use this keyword?

Answer: this keyword in Java refers to the current object in the constructor or in the method.

- When class attributes and parameterized constructors both have the same name, **this** keyword is used.
- Keywords **this** invokes the current class constructor, method of the current class, return the object of the current class, pass an argument in the constructor, and method call.

Q #21) What is the difference between Runtime and compile-time polymorphism?

Answer: Both runtime and compile-time polymorphism are two different types of polymorphism. **Their differences are explained below:**

Compile Time Polymorphism	Runtime Polymorphism
Call is resolved by a compiler in compile-time polymorphism.	Call is not resolved by the compiler in runtime polymorphism.
It is also known as static binding and method overloading.	It is also known as dynamic, late, and method overriding.
Same name methods with different parameters or methods with the same signature and different return types are compile-time polymorphism.	Same name method with the same parameters or signature associated in different classes are called method overriding.
It is achieved by function and operator overloading.	It can be achieved by pointers and virtual functions.
As all the things are executed at compile time. compile-time polymorphism is less flexible.	As things execute at run time, runtime polymorphism is more flexible.

Q #22) What object-oriented features are used in Java?

Answer: A concept of using an object in Java programming language benefits by the use of object-oriented concepts like encapsulation for binding together the state and behavior of an object, secures data access with access specifiers, features like abstraction in information hiding, inheritance to extend state, and behavior of base classes to child classes, compile-time and runtime polymorphism for method overloading and method overriding, respectively.

Q #23) What is method overloading?

Answer: When two or more methods with the same name have either a different number of parameters or different types of parameters, these methods may have or may not have different return types, then they are overloaded methods, and the feature is method overloading. Method overloading is also called compile-time polymorphism.

Q #24) What is method overriding?

Answer: When a method of sub class (derived, child class) has the same name, parameters (signature), and same return type as the method in its super class (base, parent class) then the method in the subclass is said to be overridden the method in the superclass. This feature is also known as runtime polymorphism.

Q #25) Explain constructor overloading.

Answer: More than one constructor having different parameters so that different tasks can be carried out with each constructor is known as constructor overloading. With constructor overloading, objects can be created in different ways. Various Collection classes in Java API are examples of constructor overloading.

Q #26) What types of arguments can be used in Java?

Answer: For Java methods and functions, parameter data can be sent and received in different ways. If methodB() is called from methodA(), methodA() is a caller function and methodB() is called function, arguments sent by methodA() is actual arguments and parameters of methodB() is called formal arguments.

- **Call By Value:** Changes made to formal parameter (parameters of methodB()) do not get sent back to the caller (methodA()), This method is called **call by value**. Java supports the call by value.
- **Call by Reference:** Changes made to formal parameter (parameters of methodB()) are sent back to the caller (parameters of methodB()).
- Any changes in formal parameters (parameters of methodB()) are reflected in actual parameters (arguments sent by methodA()). This is called call by reference.

Q #27) Differentiate between static and dynamic binding?

Answer: The differences between Static and Dynamic binding are explained in the below table.

Static Binding	Dynamic Binding
Static binding in Java use type of fields and class to as a resolution.	Dynamic binding in Java uses object for resolving binding.
Method Overloading is an example of static binding.	Method overriding is an example of dynamic binding.
Static binding gets resolved at compile time.	Dynamic binding gets resolved at run time.
Methods and variables using static binding are private, final and static types.	Virtual methods use dynamic binding.

Q #28) Can you explain base class, subclass, and superclass?

Answer: Base class, sub class, and super class in Java are explained as follows:

- Base class or parent class is a super class and is a class from which sub class or child class is derived.
- Sub class is a class that inherits attributes (properties) and methods (behavior) from the base class.

Q #29) Is Operator overloading supported in Java?

Answer: Operator overloading is not supported by Java as,

- It makes the interpreter put more effort to understand the actual functionality of the operator making code complex and difficult to compile.
- Operator overloading makes programs more error-prone.
- However, the feature of operator overloading can be achieved in method overloading in a simple, clear, and error-free way.

Q #30) When the finalize method is used?

Answer: finalize method is called just before the object is about to be garbage collected. This method overrides to minimize memory leaks, undertake cleanup activities by removing system resources.

Q #31) Explain about Tokens.

Answer: Tokens in the Java program are the smallest elements that the compiler recognizes. Identifiers, keywords, literals, operators, and separators are examples of tokens.

OBJECT ORIENTED PROGRAMMING

- BY TAMIL TELEGRAM TECH INTERVIEWERS

Q #1) Explain in brief what do you mean by Object Oriented Programming in Java?

Answer: OOP deals with objects, like real-life entities such as pen, mobile, bank account which has state (data) and behavior (methods).

With help of access, specifiers access to this data and methods is made secured. Concepts of encapsulation and abstraction offer data hiding and access to essentials, inheritance, and polymorphism help code reuse and overloading/overriding of methods and constructors, making applications platform-independent, secured and robust using languages like Java.

Q #2) Explain Is Java a pure Object Oriented language?

Answer: Java is not an entirely pure object-oriented programming language. **The following are the reasons:**

- Java supports and uses primitive data types such as int, float, double, char, etc.
- Primitive data types are stored as variables or on the stack instead of the heap.
- In Java, static methods can access static variables without using an object, contrary to object-oriented concepts.

Q #3) Describe class and object in Java?

Answer: Class and object play an integral role in object-oriented programming languages like Java.

- Class is a prototype or a template that has state and behavior supported by an object and used in the creation of objects.
- The object is an instance of the class, **for example,** Human is a class with the state as having a vertebral system, brain, color, and height and has behavior such as canThink(), ableToSpeak(), etc.

Q #4) What are the differences between class and objects in Java?

Answer: Following are a few major differences between class and objects in Java:

Class	Object
Class is a logical entity	Object is physical entity
Class is a template from which object can be created	Object is an instance of the class
Class is a prototype that has the state and behavior of similar objects	Objects are entities that exist in real life such as mobile, mouse, or intellectual objects such as bank account
Class is declared with class key word like class Classname { }	Object is created via new keyword as Employee emp = new Employee();
During class creation, there is no allocation of memory	During object creation, memory is allocated to the object
There is only one-way class is defined using the class keyword	Object creation can be done many ways such as using new keyword, newInstance() method, clone() and factory method.
Real-life examples of Class can be a •A recipe to prepare food. •Blue prints for an automobile engine.	Real-life examples of Object can be •A food prepared from recipe. •Engine constructed as per blue-prints.

Q #5) Why is a need for Object-oriented programming?

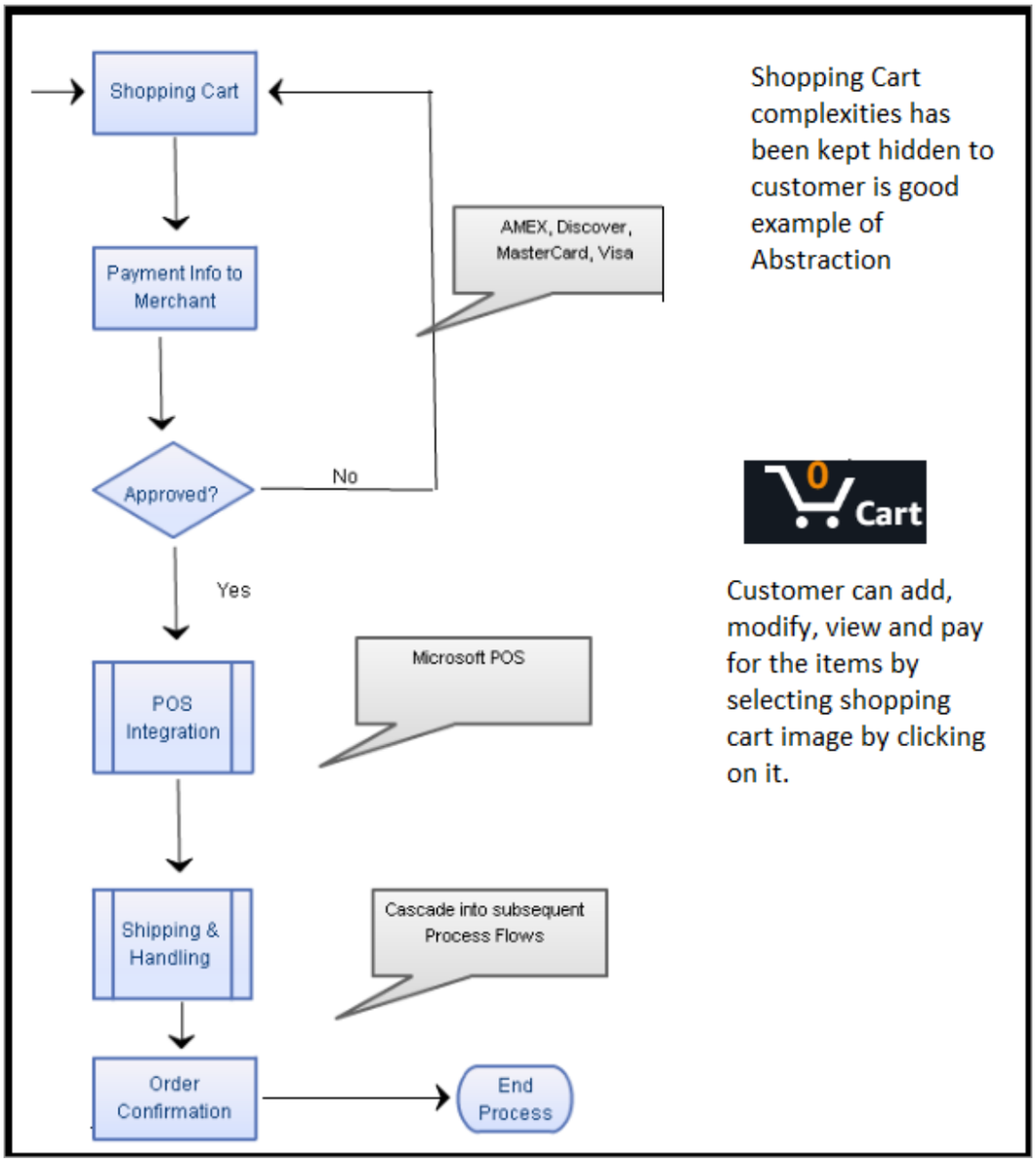
Answer: OOP provides access specifiers and data hiding features for more security and control data access, overloading can be achieved with function and operator overloading, Code Reuse is possible as already created objects in one program can be used in other programs.

Data redundancy, code maintenance, data security, and advantage of concepts such as encapsulation, abstraction, polymorphism, and inheritance in object-oriented programming provide an advantage over previously used procedural programming languages.

Q #6) Explain Abstraction with a real-time example.

Answer: Abstraction in object-oriented programming means hiding complex internals but to expose only essential characteristics and behavior with respect to context. In real life, an example of abstraction is an online shopping cart, say at any e-commerce site. Once you select a product and book order, you are just interested in receiving your product on time.

How things happen is not what you are interested in, as it is complex and kept hidden. This is known as abstraction. Similarly, take the example of ATM, the complexity of internals of how money is debited from your account is kept hidden, and you receive cash via a network. Similarly for cars, how petrol makes the engine run the automobile is extremely complex.



Q #7) Give some real-time examples and explain Inheritance.

Answer: Inheritance means one class (sub class) acquiring properties of another class (super class) by inheritance. In real life, take an example of inheritance of a normal bicycle where it is a parent class and a sports bike can be a child class, where sports bike has inherited properties and behavior of rotating wheels with pedals via gears that of a normal bike.

Q #8) How polymorphism works in Java, explain with real-life examples?

Answer: Polymorphism is an ability to have multiple forms or capability of the method to do different things. In real life, the same person performing different duties behaves differently. In-Office he is an employee, at home, he is a father, during or in after school tuitions he is a student, on weekends he plays cricket and is a player in the playground.

In Java, there are two types of polymorphism

- **Compile-time polymorphism:** This is achieved by method overloading or operator overloading.
- **Runtime polymorphism:** This is achieved by method overriding.

Q #9) How many types of inheritance are present?

Answer: Various types of inheritance are listed below:

- **Single Inheritance:** Single child class inherits characteristics of the single-parent class.
- **Multiple Inheritance:** One class inherits features of more than one base class and is not supported in Java, but the class can implement more than one interface.
- **Multilevel Inheritance:** A class can inherit from a derived class making it a base class for a new class, **for example,** a Child inherits behavior from his father, and the father has inherited characteristics from his father.
- **Hierarchical Inheritance:** One class is inherited by multiple subclasses.
- **Hybrid Inheritance:** This is a combination of single and multiple inheritances.

Q #10) What is Interface?

Answer: Interface is similar to the class where it can have methods and variables, but its methods do not have a body, just a signature known as the abstract method. Variables declared in the interface can have public, static, and final by default. Interface is used in Java for abstraction and multiple inheritances, where the class can implement multiple interfaces.

Q #11) Can you explain the advantages of Abstraction and Inheritance?

Answer: Abstraction reveals only essential details to the user and ignores or hides irrelevant or complex details. In other words, data abstraction exposes the interface and hides implementation details. Java performs abstraction with the help of interfaces and abstract classes. Advantage of abstraction is that it makes simple in viewing things by reducing or hiding the complexity of implementation.

Duplication of code is avoided, and it increases code reusability. Only essential details are revealed to the user and improves the security of the application.

Inheritance is where child class inherits functionality (behavior) of the parent class. We need not write code once written in parent class for functionality again in the child class and thus making it easier to reuse the code. The code becomes readable as well. Inheritance is used where there “is a” relation. **Example:** Hyundai **is a** car OR MS Word **is a** software.

Q #12) What is the difference between extends and implements?

Answer: Both extends and implements keyword are used for inheritance but in different ways.

The differences between Extends and Implements keywords in Java are explained below:

Extends	Implements
A class can extend another class (child extending parent by inheriting his characteristics). Interface as well inherit (using keyword extends) another interface.	A class can implement an interface
Sub class extending super class may not override all of the super class methods	Class implementing interface has to implement all the methods of the interface.
Class can only extend a single super class.	Class can implement any number of interfaces.
Interface can extend more than one interfaces.	Interface cannot implement any other interface.
Syntax: **class Child extends class Parent**	**Syntax:** **class Hybrid implements Rose**

Q #13) What are different access modifiers in Java?

Answer: Access modifiers in Java controls access scope of class, constructor, variable, method, or data member. **Various types of access modifiers are as follows:**

- **Default access modifier** is without any access specifier data members, class and methods, and are accessible within the same package.
- **Private access modifiers** are marked with the keyword private, and are accessible only within class, and not even accessible by class from the same package.
- **Protected access modifiers** can be accessible within the same package or subclasses from different packages.
- **Public access modifiers** are accessible from everywhere.

Q #14) Explain the difference between abstract class and method?

Answer: Following are some differences between abstract class and abstract method in Java:

Abstract Class	Abstract Method
Object cannot be created from the abstract class.	Abstract method has a signature but does not have a body.
Sub class created or inherit abstract class to access members of abstract class.	It is compulsory to override abstract methods of super class in their sub class.
Abstract class can contain abstract methods or non abstract methods.	Class containing abstract method should be made abstract class.

Q #15) What are the differences between method and constructor?

Answer: Following are the differences between constructors and methods in Java:

Constructors	Methods
Constructors name should match with that of Class.	Methods should not have same name as Class name.
They are used to create, initialize and allocate memory to the object.	Methods are used to execute certain statements written inside them.
Constructors are implicitly invoked by the system whenever objects are created.	Methods are invoked when it is called.
They are invoked using new keyword while creating an instance of the class (object).	Methods are invoked during program execution.
Constructor does not have return type.	Method has a return type.
Constructor cannot be inherited by the subclass.	Methods can be inherited by a sub class.

Q #16) What is a constructor in Java?

Answer: Constructor is a method without a return type and has its name the same as the class name. When we create an object, a default constructor allocates memory for an object during the compilation of Java code. Constructors are used to initializing objects and set initial values for object attributes.

Q #17) How many types of constructors can be used in Java? Please explain.

Answer: There are basically three types of constructors in Java.

These are:

1. **Default constructor:** This constructor is without any parameter and invokes every time you create an instance of a class (object). If a class is an Employee, then the syntax of the default constructor will be Employee().
2. **No-arg constructor:** As the name implies, a constructor without any argument is called a no-arg constructor.
3. **Parameterized constructor:** Constructor with a number of parameters is called a parameterized constructor. You are required to provide arguments, i.e. initial values with respect to the data type of parameters in that constructor.

Q #18) Why new keyword is used in Java?

Answer: When we create an instance of class, i.e. objects, we use the Java keyword **new**. It allocates memory in the heap area where JVM reserve space for an object. Internally, it invokes the default constructor as well.

Syntax:

```
Class_name obj = new Class_name();
```

Q #19) When do you use the super keyword?

Answer: Super is a Java keyword used to identify or refer parent (base) class.

- We can use super to access super class constructor and call methods of the super class.
- When method names are the same in super class and sub class, to refer super class, the **super** keyword is used.
- To access the same name data members of parent class when they are present in parent and child class.
- **Super** can be used to make an explicit call to no-arg and parameterized constructors of the parent class.
- Parent class method access can be done using **super**, when child class has method overridden.

Q #20) When do you use this keyword?

Answer: this keyword in Java refers to the current object in the constructor or in the method.

- When class attributes and parameterized constructors both have the same name, **this** keyword is used.
- Keywords **this** invokes the current class constructor, method of the current class, return the object of the current class, pass an argument in the constructor, and method call.

Q #21) What is the difference between Runtime and compile-time polymorphism?

Answer: Both runtime and compile-time polymorphism are two different types of polymorphism. **Their differences are explained below:**

Compile Time Polymorphism	Runtime Polymorphism
Call is resolved by a compiler in compile-time polymorphism.	Call is not resolved by the compiler in runtime polymorphism.
It is also known as static binding and method overloading.	It is also known as dynamic, late, and method overriding.
Same name methods with different parameters or methods with the same signature and different return types are compile-time polymorphism.	Same name method with the same parameters or signature associated in different classes are called method overriding.
It is achieved by function and operator overloading.	It can be achieved by pointers and virtual functions.
As all the things are executed at compile time. compile-time polymorphism is less flexible.	As things execute at run time, runtime polymorphism is more flexible.

Q #22) What object-oriented features are used in Java?

Answer: A concept of using an object in Java programming language benefits by the use of object-oriented concepts like encapsulation for binding together the state and behavior of an object, secures data access with access specifiers, features like abstraction in information hiding, inheritance to extend state, and behavior of base classes to child classes, compile-time and runtime polymorphism for method overloading and method overriding, respectively.

Q #23) What is method overloading?

Answer: When two or more methods with the same name have either a different number of parameters or different types of parameters, these methods may have or may not have different return types, then they are overloaded methods, and the feature is method overloading. Method overloading is also called compile-time polymorphism.

Q #24) What is method overriding?

Answer: When a method of sub class (derived, child class) has the same name, parameters (signature), and same return type as the method in its super class (base, parent class) then the method in the subclass is said to be overridden the method in the superclass. This feature is also known as runtime polymorphism.

Q #25) Explain constructor overloading.

Answer: More than one constructor having different parameters so that different tasks can be carried out with each constructor is known as constructor overloading. With constructor overloading, objects can be created in different ways. Various Collection classes in Java API are examples of constructor overloading.

Q #26) What types of arguments can be used in Java?

Answer: For Java methods and functions, parameter data can be sent and received in different ways. If methodB() is called from methodA(), methodA() is a caller function and methodB() is called function, arguments sent by methodA() is actual arguments and parameters of methodB() is called formal arguments.

- **Call By Value:** Changes made to formal parameter (parameters of methodB()) do not get sent back to the caller (methodA()), This method is called **call by value**. Java supports the call by value.
- **Call by Reference:** Changes made to formal parameter (parameters of methodB()) are sent back to the caller (parameters of methodB()).
- Any changes in formal parameters (parameters of methodB()) are reflected in actual parameters (arguments sent by methodA()). This is called call by reference.

Q #27) Differentiate between static and dynamic binding?

Answer: The differences between Static and Dynamic binding are explained in the below table.

Static Binding	Dynamic Binding
Static binding in Java use type of fields and class to as a resolution.	Dynamic binding in Java uses object for resolving binding.
Method Overloading is an example of static binding.	Method overriding is an example of dynamic binding.
Static binding gets resolved at compile time.	Dynamic binding gets resolved at run time.
Methods and variables using static binding are private, final and static types.	Virtual methods use dynamic binding.

Q #28) Can you explain base class, subclass, and superclass?

Answer: Base class, sub class, and super class in Java are explained as follows:

- Base class or parent class is a super class and is a class from which sub class or child class is derived.
- Sub class is a class that inherits attributes (properties) and methods (behavior) from the base class.

Q #29) Is Operator overloading supported in Java?

Answer: Operator overloading is not supported by Java as,

- It makes the interpreter put more effort to understand the actual functionality of the operator making code complex and difficult to compile.
- Operator overloading makes programs more error-prone.
- However, the feature of operator overloading can be achieved in method overloading in a simple, clear, and error-free way.

Q #30) When the finalize method is used?

Answer: finalize method is called just before the object is about to be garbage collected. This method overrides to minimize memory leaks, undertake cleanup activities by removing system resources.

Q #31) Explain about Tokens.

Answer: Tokens in the Java program are the smallest elements that the compiler recognizes. Identifiers, keywords, literals, operators, and separators are examples of tokens.