BASIC TECHNICAL QUESTIONS

Depending on the type of role you are applying for the interview questions can differ. It is common for interviewers to ask behavioural type questions that start with "tell me about a time when..." and will have a follow up to gauge your leadership, problem-solving, and teamwork skills.

Core business activities determine an organisation’s talent requirements. A candidate’s technical ability and understanding of the business will determine their ability to perform the job. You can assess the candidate’s basic knowledge of the subject, [problem-solving skills](https://www.merittrac.com/cognitive-tests/problem-solving-assessment), coding skills, and other requisites through an online test. The type of technical skills assessed during these tests may vary depending on the job role.

1. Do you have any technical certifications?
2. How many programming languages do you know?
3. What are the different types of OS you are comfortable working with?
4. What is the extent of your technical expertise?
5. How many development tools have you used?
6. What do you do to improve your technical skills?
7. What are the technical websites you follow?
8. Why is a solution design document important?
9. Have you used any source code tools? If yes, explain them.
10. What do you like most about the IT industry? What do you enjoy the least about it?
11. Give an example of how you apply your technical knowledge in a practical way?
12. If you have to change your career path, what job would you prefer?
13. Whenever you solve a problem, who do you keep in mind? The end-user, the business, or yourself, and why?
14. What is the importance of learning on the job, and how do you learn new technologies?
15. Do you reuse code? If yes, how do you do it efficiently?
16. How do you ensure the quality of your deliverables?
17. Which technical skill do you wish to possess?
18. What was the recent technical project you worked on? What were your key responsibilities?
19. What is the production deployment process you follow?
20. How will you use your technical knowledge/expertise to perform this job if you are selected?
21. **1. What is Artificial Intelligence?**

Ans. – [Artificial Intelligence](https://www.monsterindia.com/career-advice/5-artificial-intelligence-jobs-in-demand-and-how-to-get-them-7669.html) is a technology that is used in [computer science](https://www.monsterindia.com/career-advice/branch-in-focus-computer-science-it/). It is used for creating an intelligent machine that can behave like a human; it can solve any task.

1. **2. Do you have any idea why programming languages exist?**

**Ans**. – Yes, the [programming language](https://www.monsterindia.com/career-advice/5-programming-languages-that-every-techie-should-master/) is a formal language that can produce various information by coding. It is the way that programmers used to communicate with computers.

1. **What is the full form of SDLC?**

Ans. – The full form of SDLC is [Software Development](https://www.monsterindia.com/career-advice/software-testing-vs-software-development-which-job-you-should-go-for/) Life Cycle. It is a well-known process that provides quality software products in a short time.

1. **Name some popular operating systems.**

Ans. – [Microsoft](https://www.monsterindia.com/search/microsoft-jobs), [OSX](https://www.monsterindia.com/search/osx-jobs), [Linux](https://www.monsterindia.com/career-advice/9-questions-you-should-study-before-your-next-linux-interview-7881.html), and Windows are some popular operating systems.

### ****25. If you have to learn a new programming language for this job, what will you do?****

Ans. – I’m always ready to learn. And I’ve learnt many programming languages like- [C](https://www.monsterindia.com/career-advice/10-important-questions-to-help-you-ace-your-c-programming-interview-7858.html), C++, [C#](https://www.monsterindia.com/career-advice/top-8-c-interview-questions-and-answers-for-experience-professionals-7859.html), [Java](https://www.monsterindia.com/career-advice/top-20-most-frequently-asked-java-interview-questions-7831.html), [JavaScript](https://www.monsterindia.com/career-advice/javascript-interview-questions-you-should-know-about-before-your-next-interview-7944.html), [SQL](https://www.monsterindia.com/career-advice/20-common-sql-interview-questions-answers-7945.html), [Python](https://www.monsterindia.com/career-advice/10-python-questions-answers-you-must-read-before-your-next-interview-7875.html), and Ruby. So, I think I will learn the new language smoothly as well.

### ****26. What interests pulled you up about this job?****

Ans. – I’m highly interested in joining as a programmer in this job. I think my master’s programs are helping me to grow in this field, and I can implement my skills also.

### Interview Questions about education –

This is also an important part of an interview. The interviewer gets some idea from this section of the interview about how much the candidate knows about technical terms, what kind of training they need etc.

### ****27. What is your educational background?****

Ans. I have done my [B.Sc. Degree in Computer Science](https://www.monsterindia.com/search/bsc-computer-science-jobs) from XYZ college. I have also pursued a master’s degree in Python and Java.

### ****28. Why should we hire you?****

Ans. – I am a fresher but I have the required qualification that you are looking for. I cannot ask you to hire me, but I can assure you that I will prove my worth and deliver quality work if you hire me.

### ****29. In what position do you see yourself 5 years from now?****

Ans. – As a fresher, it is very difficult to say but still, I want to be in a successful position with lots of experience and give the proud moment for my company.

### ****30. If you have to learn a new programming language for this job, what will you do?****

Ans. – I’m always ready to learn. And I’ve learnt many programming languages like- [C](https://www.monsterindia.com/career-advice/10-important-questions-to-help-you-ace-your-c-programming-interview-7858.html), C++, [C#](https://www.monsterindia.com/career-advice/top-8-c-interview-questions-and-answers-for-experience-professionals-7859.html), [Java](https://www.monsterindia.com/career-advice/top-20-most-frequently-asked-java-interview-questions-7831.html), [JavaScript](https://www.monsterindia.com/career-advice/javascript-interview-questions-you-should-know-about-before-your-next-interview-7944.html), [SQL](https://www.monsterindia.com/career-advice/20-common-sql-interview-questions-answers-7945.html), [Python](https://www.monsterindia.com/career-advice/10-python-questions-answers-you-must-read-before-your-next-interview-7875.html), and Ruby. So, I think I will learn the new language smoothly as well.

### ****31. What interests pulled you up about this job?****

Ans. – I’m highly interested in joining as a programmer in this job. I think my master’s programs are helping me to grow in this field, and I can implement my skills also.

### Interview Questions about education –

This is also an important part of an interview. The interviewer gets some idea from this section of the interview about how much the candidate knows about technical terms, what kind of training they need etc.

### ****1. What is your educational background?****

Ans. I have done my [B.Sc. Degree in Computer Science](https://www.monsterindia.com/search/bsc-computer-science-jobs) from XYZ college. I have also pursued a master’s degree in Python and Java.

### ****2. Why should we hire you?****

Ans. – I am a fresher but I have the required qualification that you are looking for. I cannot ask you to hire me, but I can assure you that I will prove my worth and deliver quality work if you hire me.

### ****3. In what position do you see yourself 5 years from now?****

Ans. – As a fresher, it is very difficult to say but still, I want to be in a successful position with lots of experience and give the proud moment for my company.

### 4. What is your troubleshooting process?

**What they’re really asking:** How do you solve a problem?

Troubleshooting is an essential part of an IT support role, so a question like this is likely to come up in an interview in some form.

To answer, work your way through troubleshooting steps. This can be something like: understanding and identifying the problem, determining a cause, testing a solution, ensuring the problem is resolved afterwards, and ending with documenting your findings.

**Other forms this question might take:**

* Somebody has come to you with Wifi issues. How would you resolve this?
* Tell me about a problem you've solved.

### 5. What is an IP address?

**What they’re really asking:** Can you talk about technical concepts in easy-to-understand language?

With this type of question, an interviewer might be looking for a straightforward answer to see if you know basic IT terminology. But this can also be a way to see how well you explain technical concepts—a key part of your job, especially if you’re applying for a help desk position.

Think about the main function of the item in question. What does it help people accomplish? Why is it important? You should be able to define the item, and speak to the broader context of why it matters.

**Other forms this question might take:**

* What is a CPU?
* What is a VPN?
* What is DNS?

### 3. How would you move a file using command prompt (or command line)?

**What they’re really asking:** Do you have technical skills, or can you troubleshoot an unknown issue on the spot?

This is a specific question, and it’s great if you know how to complete this task off the top of your head. But a technical question you might not know how to answer right away can be a way for an interviewer to see how you think on your feet.

If you know how to do the task, that’s perfect. If you don’t, try walking the interviewer through a process you would take to figure out how to complete it.

### 6. How do you stay on top of new technology?

**What they’re really asking:** Are you seeking to learn and grow?

Because technology is an ever-evolving field, it’s important for IT professionals to stay on top of new innovations that can improve a workplace (or pose new security threats). An interviewer can ask this question to see how plugged-in you are to the IT community.

Maybe you follow tech experts or companies on social media, or receive their newsletters. Maybe you’ve taken courses that incorporate newer technologies into their curriculum, or belong to a professional organization. Whatever your response, try to show your interest in new technology and willingness to stay on top of new innovations.

**Other forms this question might take:**

* Which industry leaders do you enjoy following?
* What podcasts or blogs do you follow?

### 7. How familiar are you with different operating systems?

**What they’re really asking:** Are you versatile?

Different companies might use Windows, Mac, or Linux operating systems. Some jobs may want to hire somebody that is familiar with multiple operating systems, while others might require you to know only one.

Check the job description to see if they indicate a preference for any particular operating system. In the interview, be honest about what you’ve worked with before and how familiar you are with others. Displaying a willingness to learn always helps. You can also take a course to learn the basics of ones you don’t know as well—Coursera offers some on [Windows](https://www.coursera.org/learn/windows-server-management-security) or [Linux server management and security](https://www.coursera.org/learn/linux-server-management-security).

**Other forms this question might take:**

* Which operating systems have you used?
* What is your favorite operating system to work with?

### 8. How would you make sure a computer network is secure?

**What they’re really asking:** Do you have the skills to keep our data safe?

This is a question that helps to gauge your technical knowledge in addition to showing an interviewer how you would approach a problem. Basic security knowledge should be a part of your arsenal.

Walk through the basic security features you would implement—firewalls, routers, VPNs. Think also about good security practices you can implement, like creating strong passwords and keeping software updated. If you have any past experiences at work or in your private life in which you dealt with security issues, this can be a good time to share them.

**Other forms this question might take:**

* Have you ever experienced a security breach?
* How do you use firewalls?

### 9. Why did you decide to go into IT?

**What they’re really asking:** Why are you here?

While technical skills can get you far, there's a lot in IT work that can be learned on the job. Because of this, employers might look for somebody who has other qualities that can be linked to success, like passion and curiosity. This question can also be a way for employers to get to know you and your story.

Reflect on what aspect of IT you like. Is it solving problems for people, working with computers, learning new things, or something else? Showing your motivations for being in IT can give hiring managers a better idea of who you are, and your enthusiasm for the job and field.

**Other forms this question might take:**

* Tell me about yourself.
* Why are you interested in this job?

### 10. Why should we hire you?

**What they’re really asking:** What makes you uniquely qualified for this position?

An interviewer will likely ask general questions to assess your interest in the job and your capabilities. Hearing directly from you about what you consider your strongest assets can be a good way for employers to assess your motivations, your skill set, and your fit for the job.

Reiterate your strengths and skills that are applicable to the job. Show how your qualities line up with what they’re looking for, both in terms of technical skill and personality.

It can also help to convey enthusiasm, and show that you’re willing and eager to learn. If you’re genuinely excited for the job and the company, that can show through. Be sure to find a few points in the job description or company website that makes you eager to work for them before the interview.

**Other forms this question might take:**

* What makes you the right person for this role?
* What will you bring to our team?

DATABASE MANAGEMENT SYSTEM:

**Q #1) What is** **DBMS used for?**

**Answer:** DBMS, commonly known as Database Management System, is an application system whose main purpose revolves around the **data**. This is a system that allows its user to store the data, define it, retrieve it and update the information about the data inside the database.

**Q #2) What is meant by a Database?**

* **Answer:** In simple terms, Database is a collection of data in some organized way to required.
* **Enforcement of Integrity Constraints:**Integrity Constraints are very important to be enforced on the data so that the refined data after putting some constraints are stored in the database and this is followed by DBMS.
* **Independence of data:** It simply means that you can change the structure of the data without affecting the structure of any of the application programs.

**Q #4) What is the purpose of normalization in DBMS?**

**Answer:** Normalization is the process of analyzing the relational schemas which are based on their respective functional dependencies and the primary keys in order to fulfill certain properties.

**The properties include:**

* To minimize the redundancy of the data.
* To minimize the Insert, Delete and Update Anomalies.

**Q #5) What are the different types of languages that are available in the DBMS?**

**Answer:** Basically, there are 3 types of languages in the DBMS as mentioned below:

* **DDL:**DDL is **Data Definition Language** which is used to define the database and schema structure by using some set of SQL Queries like **CREATE**, **ALTER**, **TRUNCATE**, **DROP** and **RENAME.**
* **DCL:**DCL is **Data Control Language** which is used to control the access of the users inside the database by using some set of SQL Queries like **GRANT** and **REVOKE.**
* **DML:**DML is **Data Manipulation Language** which is used to do some manipulations in the database like Insertion, Deletion, etc. by using some set of SQL Queries like **SELECT**, **INSERT**, **DELETE** and **UPDATE.**

**Q #6) What is the purpose of SQL?**

**Answer:** SQL stands for **Structured Query Language** whose main purpose is to interact with the relational databases in the form of inserting and updating/modifying the data in the database.

**Q #7) Explain the concepts of a Primary key and Foreign Key.**

**Answer: Primary Key** is used to uniquely identify the records in a database table while **Foreign Key**is mainly used to link two or more tables together, as this is a particular field(s) in one of the database tables which are the primary key of some other table.

**Example**: There are 2 tables – Employee and Department. Both have one common field/column as ‘**ID’** where ID is the primary key of the **Employee** table while this is the foreign key for the **Department** table.

**Q #8) What are the main differences between Primary key and Unique Key?**

**Answer: Given below are few differences:**

* The main difference between the Primary key and Unique key is that the Primary key can never have a null value while the Unique key may consist of null value.
* In each table, there can be only one primary key while there can be more than one unique key in a table.

**Q #9) What is the concept of sub-query in terms of SQL?**

**Answer:** Sub-query is basically the query which is included inside some other query and can also be called as an inner query which is found inside the outer query.

**Q #10) What is the use of DROP command and what are the differences between DROP, TRUNCATE and DELETE commands?**

**Answer: DROP** command is a DDL command which is used to drop/delete the existing table, database, index or view from the database.

**The major difference between DROP, TRUNCATE and DELETE commands are:**

**DROP** and **TRUNCATE** commands are the **DDL** commands which are used to delete tables from the database and once the table gets deleted, all the privileges and indexes and so should be used only when necessary.

**DELETE** command, on the other hand, is a **DML** Command which is also used to delete rows from the table and this can be rolled back.

**Note**: It is recommended to use the ‘WHERE’ clause along with the DELETE command else the complete table will get deleted from the database.

**Q #11) What is the main difference between UNION and UNION ALL?**

**Answer:** UNION and UNION ALL are used to join the data from 2 or more tables but UNION removes duplicate rows and picks the rows which are distinct after combining the data from the tables whereas UNION ALL does not remove the duplicate rows, it just picks all the data from the tables.

**Q #12) Explain the concept of ACID properties in DBMS?**

**Answer:** ACID properties is the combination of Atomicity, Consistency, Isolation, and Durability properties. These properties are very helpful in allowing a safe and secure way of sharing the data among multiple users.

* **Atomicity**: This is based on the concept of “either all or nothing” which basically means that if any update occurs inside the database then that update should either be available to all the others beyond user and application program or it should not be available to anyone beyond the user and application program.
* **Consistency**: This ensures that the consistency is maintained in the database before or after any transaction that takes place inside the database.
* **Isolation**: As the name itself suggests, this property states that each transaction that occurs is in isolation with others i.e. a transaction which has started but not yet completed should be in isolation with others so that the other transaction does not get impacted with this transaction.
* **Durability**: This property states that the data should always be in a durable state i.e. any data which is in the committed state should be available in the same state even if any failure or restart occurs in the system.

**Q #13) What is Correlated Subquery in DBMS?**

**Answer:** A Subquery is also known as a nested query i.e. a query written inside some query. When a Subquery is executed for each of the rows of the outer query then it is termed as a Correlated Subquery.

**Example of Non-Correlated Subquery is:**

|  |
| --- |
| SELECT \* from EMP WHERE ‘RIYA’ IN (SELECT Name from DEPT WHERE EMP.EMPID=DEPT.EMPID); |

Here, the inner query is not executed for each of the rows of the outer query.

**Q #14) Explain Entity, Entity Type, and Entity Set in DBMS?**

**Answer:**

**Entity** is an object, place or thing which has its independent existence in the real world and about which data can be stored in a database. **For Example,** any person, book, etc.

**Entity Type**is a collection of entities that have the same attributes. **For Example,** the STUDENT table contains rows in which each row is an entity holding the attributes like name, age, and id of the students, hence STUDENT is an Entity Type which holds the entities having the same attributes.

**Entity Set**is a collection of entities of the same type. **For Example,** A collection of the employees of a firm.

**Q #15) What are the different levels of abstraction in the DBMS?**

**Answer:**There are 3 levels of data abstraction in the DBMS.

**They include:**

* **Physical Level:**This is the lowest level of the data abstraction which states how the data is stored in the database.
* **Logical Level:**This is the next level of the data abstraction which states the type of the data and the relationship among the data that is stored in the database.
* **View Level:** This is the highest level in the data abstraction which shows/states only a part of the database.

**Q #16) What integrity rules exist in the DBMS?**

**Answer:** There are 2 major integrity rules that exist in the DBMS.

**They are:**

* **Entity Integrity:** This states a very important rule that value of a Primary key can never have a NULL value.
* **Referential Integrity:**This rule is related to the Foreign key which states that either the value of a Foreign key is a NULL value or it should be the primary key of any other relation.

**Q #17) What is E-R model in the DBMS?**

**Answer:** E-R model is known as an **Entity-Relationship model** in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

**Q #18) What is a functional dependency in the DBMS?**

**Answer:** This is basically a constraint which is useful in describing the relationship among the different attributes in a relation.

**Example:** If there is some relation ‘R1’ which has 2 attributes as Y and Z then the functional dependency among these 2 attributes can be shown as **Y->Z** which states that Z is functionally dependent on Y.

**Q #19) What is 1NF in the DBMS?**

**Answer:** 1NF is known as the **First Normal Form**.

This is the easiest form of the normalization process which states that the **d**omain of an attribute should have only atomic values. The objective of this is to remove the duplicate columns that are present in the table.

**Q #20) What is 2NF in the DBMS?**

**Answer:**2NF is the **Second Normal Form**.

Any table is said to have in the 2NF if it satisfies the following 2 conditions:

* A table is in the 1NF.
* Each non-prime attribute of a table is said to be functionally dependent in totality on the primary key.

**Q #21) What is 3NF in the DBMS?**

**Answer:** 3NF is the **Third Normal Form.**

Any table is said to have in the 3NF if it satisfies the following 2 conditions:

* A table is in the 2NF.
* Each non-prime attribute of a table is said to be non-transitively dependent on every key of the table.

**Q #22) What is BCNF in the DBMS?**

**Answer:** BCNF is the **Boyce Codd Normal Form**which is stricter than the 3NF**.**

Any table is said to have in the BCNF if it satisfies the following 2 conditions:

* A table is in the 3NF.
* For each of the functional dependency X->Y that exists, X is the super key of a table.

**Q #23) What is a CLAUSE in terms of SQL?**

**Answer:** This is used with the SQL queries to fetch specific data as per the requirements on the basis of the conditions that are put in the SQL. This is very helpful in picking the selective records from the complete set of the records.

**For Example,** There is a query which has WHERE condition or the query with the HAVING clause.

**Q #24) How can you get the alternate records from the table in the SQL?**

**Answer:** **If you want to fetch the odd numbers then the following query can be used:**

|  |
| --- |
| SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=1; |

**If you want to fetch the even numbers, then the following query can be used:**

|  |
| --- |
| SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=0; |

**Q #25) How is the pattern matching done in the SQL?**

**Answer:** With the help of the LIKE operator, pattern matching is possible in the SQL.’**%**’ is used with the LIKE operator when it matches with the 0 or more characters and **‘\_**’ is used to match the one particular character.

**Example:**

|  |
| --- |
| SELECT \* from Emp WHERE name like ‘b%’; |
| SELECT \* from Emp WHERE name like ‘hans\_’; |

**Q #26) What is a join in the SQL?**

**Answer:** A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

**Q #27) What are different types of joins in SQL?**

**Answer: There are 4 types of SQL Joins:**

* **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.
* **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
* **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
* **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

**Q #28) What is meant by trigger?**

**Answer:**Trigger is one of the very important codes or programs which get executed **automatically in response to the events** that occur in a table or a view. **For Example,** If a new record is inserted in an employee database then the data gets created automatically in the related tables like salary, department and roles tables.

**Q #29) Explain the Stored Procedure.**

**Answer:** Stored Procedure is a group of SQL statements in the form of a function that has some unique name and is stored in relational database management systems(RDBMS) and can be accessed whenever required.

**Q #30) What is RDBMS?**

**Answer:** RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

**Q #31) What are the different type of relationships in the DBMS?**

**Answer: Relationships in DBMS depicts an association between the tables.**

**Different types of relationships are:**

* **One-to-One:** This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables. **Example:** Among a married couple, both wife and husband can have only one spouse.
* **One-to-Many:** This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one or none records in the related table. **Example:** A Mother can have many children.
* **Many-to-Many:**This states that both the tables can be related to many other tables. **Example:** One can have many siblings and so do they have.

facilitate its user’s to easily access, manage and upload the data.

**Q #3) Why is the use of DBMS recommended? Explain by listing some of its major advantages.**

**Answer:** **Some of the major advantages of DBMS are as follows:**

* **Controlled Redundancy:** DBMS supports a mechanism to control the redundancy of data inside the database by integrating all the data into a single database and as data is stored at only one place, the duplicity of data does not happen.
* **Data Sharing:**Sharing of data among multiple users simultaneously can also be done in DBMS as the same database will be shared among all the users and by different application programs.
* **Backup and Recovery Facility:**DBMS minimizes the pain of creating the backup of data again and again by providing a feature of ‘backup and recovery’ which automatically creates the data backup and restores the data whenever

### How do you communicate with an RDBMS?

You have to use Structured Query Language (SQL) to communicate with the RDBMS. Using queries of SQL, we can give the input to the database and then after processing of the queries database will provide us the required output.

### 49) What is the 3-Tier architecture?

The 3-Tier architecture contains another layer between the client and server. Introduction of 3-tier architecture is for the ease of the users as it provides the GUI, which, make the system secure and much more accessible. In this architecture, the application on the client-end interacts with an application on the server which further communicates with the database system.

 What is the difference between a DELETE command and TRUNCATE command?

**DELETE command**: DELETE command is used to delete rows from a table based on the condition that we provide in a WHERE clause.

* DELETE command delete only those rows which are specified with the WHERE clause.
* DELETE command can be rolled back.
* DELETE command maintain a log, that's why it is slow.
* DELETE use row lock while performing DELETE function.

**TRUNCATE command**: TRUNCATE command is used to remove all rows (complete data) from a table. It is similar to the DELETE command with no WHERE clause.

* The TRUNCATE command removes all the rows from the table.
* The TRUNCATE command cannot be rolled back.
* The TRUNCATE command doesn't maintain a log. That's why it is fast.
* TRUNCATE use table log while performing the TRUNCATE function.

### What is stored procedure?

A stored procedure is a group of SQL statements that have been created and stored in the database. The stored procedure increases the reusability as here the code or the procedure is stored into the system and used again and again that makes the work easy, takes less time in processing and decreases the complexity of the system. So, if you have a code which you need to use again and again then save that code and call that code whenever it is required.

### Explain ACID properties

ACID properties are some basic rules, which has to be satisfied by every transaction to preserve the integrity. These properties and rules are:

**ATOMICITY:** Atomicity is more generally known as ?all or nothing rule.' Which implies all are considered as one unit, and they either run to completion or not executed at all.

**CONSISTENCY:** This property refers to the uniformity of the data. Consistency implies that the database is consistent before and after the transaction.

**ISOLATION:** This property states that the number of the transaction can be executed concurrently without leading to the inconsistency of the database state.

**DURABILITY:** This property ensures that once the transaction is committed it will be stored in the non-volatile memory and system crash can also not affect it anymore.

2) What is 2NF?

**2NF** is the **Second Normal Form**. A table is said to be 2NF if it follows the following conditions:

* The table is in 1NF, i.e., firstly it is necessary that the table should follow the rules of 1NF.
* Every non-prime attribute is fully functionally dependent on the primary key, i.e., every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted, then even the non\_key element will still be saved in the database.

43) What is 3NF?

**3NF** stands for **Third Normal Form**. A database is called in 3NF if it satisfies the following conditions:

* It is in second normal form.
* There is no transitive functional dependency.
* For example: X->Z

**Where:**  
X->Y  
Y does not -> X  
Y->Z so, X->Z

44) What is BCNF?

**BCMF** stands for **Boyce-Codd Normal Form**. It is an advanced version of 3NF, so it is also referred to as 3.5NF. BCNF is stricter than 3NF.

A table complies with BCNF if it satisfies the following conditions:

* It is in 3NF.
* For every functional dependency X->Y, X should be the super key of the table. It merely means that X cannot be a non-prime attribute if Y is a prime attribute.

COMPUTER NETWORKS:

**What is the network?**

According to Merriam Webster, Network is usually an **informally interconnected group** or association of different entities like a person, computers, radio stations, etc.

For example, Dominos has a network of 1232 branches across India. As the name suggests the computer network is a system of peripherals or computers interconnected with each other and has a standard communication channel established between them to exchange different types of information and data.

**Why is the computer network so important?**

Have you ever heard of the Internet or NET? I guess you have, as you are already reading this article on Interviewbit surfing through the internet. But, have you ever thought about the internet? The Internet is a network of a network connecting all different network-enabled devices which enable data and information sharing between them and that makes computer networks a core part of our life and technical interviews.

### 1. How are Network types classified?

Network types can be classified and divided based on the area of distribution of the network. The below diagram would help to understand the same:

Network Types

### 2. Explain different types of networks.

Below are few types of networks:

| **Type** | **Description** |
| --- | --- |
| PAN (Personal Area Network) | Let devices connect and communicate over the range of a person. E.g. connecting Bluetooth devices. |
| LAN (Local Area Network) | It is a privately owned network that operates within and nearby a single building like a home, office, or factory |
| MAN (Metropolitan Area Network) | It connects and covers the whole city. E.g. TV Cable connection over the city |
| WAN (Wide Area Network) | It spans a large geographical area, often a country or continent. The Internet is the largest WAN |
| GAN (Global Area Network) | It is also known as the Internet which connects the globe using satellites. The Internet is also called the Network of WANs. |

### 3. Explain LAN (Local Area Network)

### LANs are widely used to connect computers/laptops and consumer electronics which enables them to share resources (e.g., printers, fax machines) and exchange information. When LANs are used by companies or organizations, they are called **enterprise networks**. There are two different types of LAN networks i.e. wireless LAN (no wires involved achieved using Wi-Fi) and wired LAN (achieved using LAN cable). Wireless LANs are very popular these days for places where installing wire is difficult. The below diagrams explain both wireless and wired LAN.

### 4. Tell me something about VPN (Virtual Private Network)

VPN or the Virtual Private Network is a private WAN (Wide Area Network) built on the internet. It allows the creation of a secured tunnel (protected network) between different networks using the internet (public network). By using the VPN, a client can connect to the organization’s network remotely. The below diagram shows an organizational WAN network over Australia created using VPN:

VPN (Virtual Private Network)

### 5. What are the advantages of using a VPN?

Below are few advantages of using VPN:

* VPN is used to connect offices in different geographical locations remotely and is cheaper when compared to WAN connections.
* VPN is used for secure transactions and confidential data transfer between multiple offices located in different geographical locations.
* VPN keeps an organization’s information secured against any potential threats or intrusions by using virtualization.
* VPN encrypts the internet traffic and disguises the online identity.

### 6. What are the different types of VPN?

Few types of VPN are:

* **Access VPN:** Access VPN is used to provide connectivity to remote mobile users and telecommuters. It serves as an alternative to dial-up connections or ISDN (Integrated Services Digital Network) connections. It is a low-cost solution and provides a wide range of connectivity.
* **Site-to-Site VPN:** A Site-to-Site or Router-to-Router VPN is commonly used in large companies having branches in different locations to connect the network of one office to another in different locations. There are 2 sub-categories as mentioned below:
* **Intranet VPN:** Intranet VPN is useful for connecting remote offices in different geographical locations using shared infrastructure (internet connectivity and servers) with the same accessibility policies as a private WAN (wide area network).
* **Extranet VPN:** Extranet VPN uses shared infrastructure over an intranet, suppliers, customers, partners, and other entities and connects them using dedicated connections.

### 7. What are nodes and links?

**Node:** Any communicating device in a network is called a Node. Node is the point of intersection in a network. It can send/receive data and information within a network. Examples of the node can be computers, laptops, printers, servers, modems, etc.

**Link:** A link or edge refers to the connectivity between two nodes in the network. It includes the type of connectivity (wired or wireless) between the nodes and protocols used for one node to be able to communicate with the other.

Nodes and Links

### 8. What is the network topology?

Network topology is a physical layout of the network, connecting the different nodes using the links. It depicts the connectivity between the computers, devices, cables, etc.

### 9. Define different types of network topology

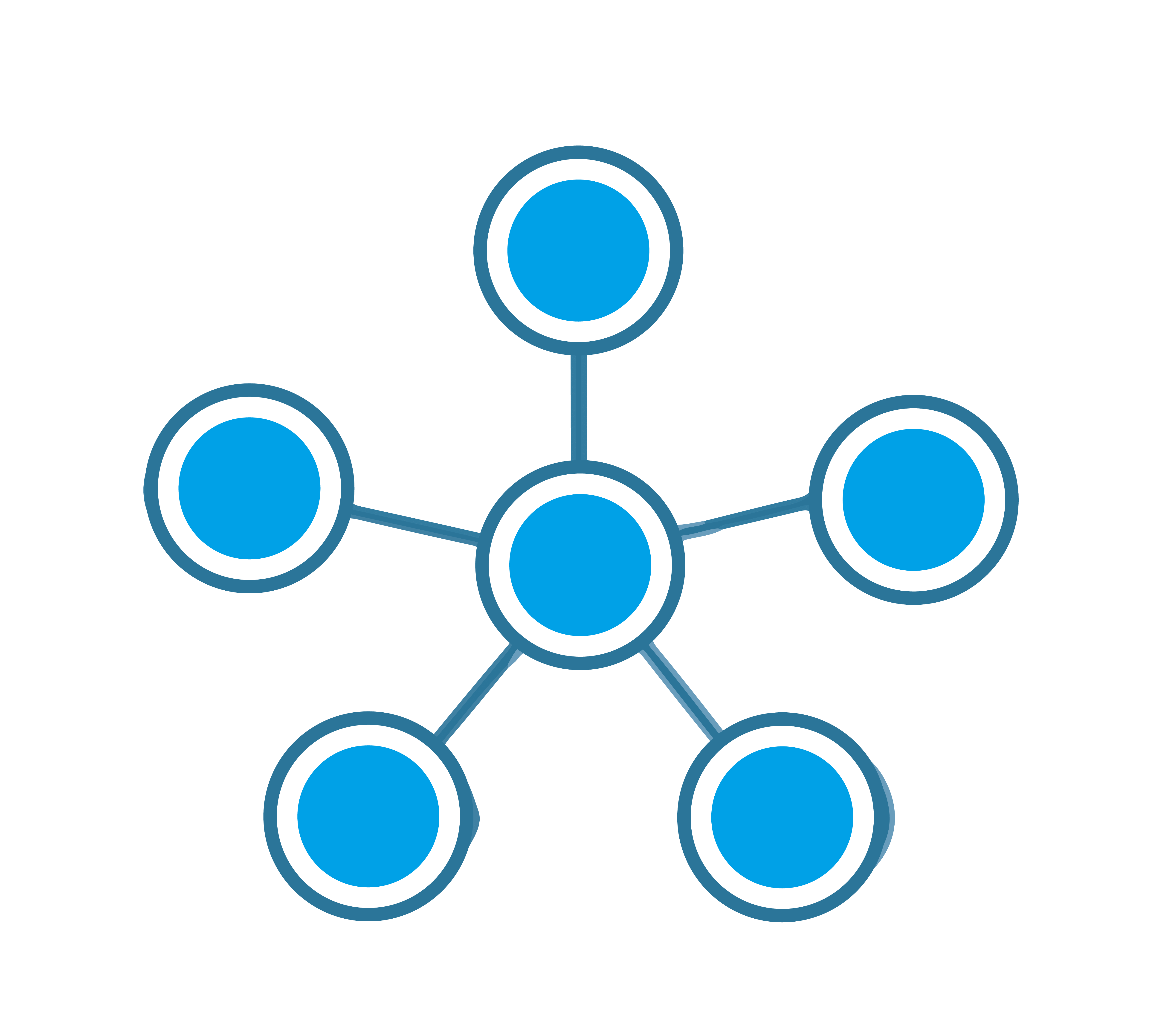
The different types of network topology are given below:

**Bus Topology:**

Bus Topology

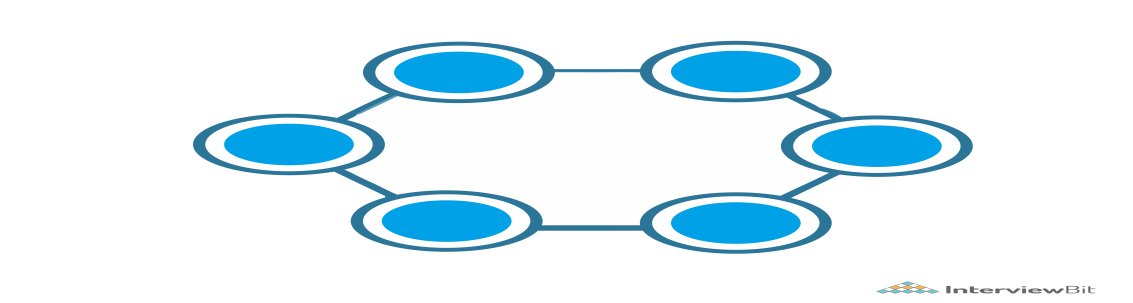
* All the nodes are connected using the central link known as the bus.
* It is useful to connect a smaller number of devices.
* If the main cable gets damaged, it will damage the whole network.

**Star Topology:**

Star Topology

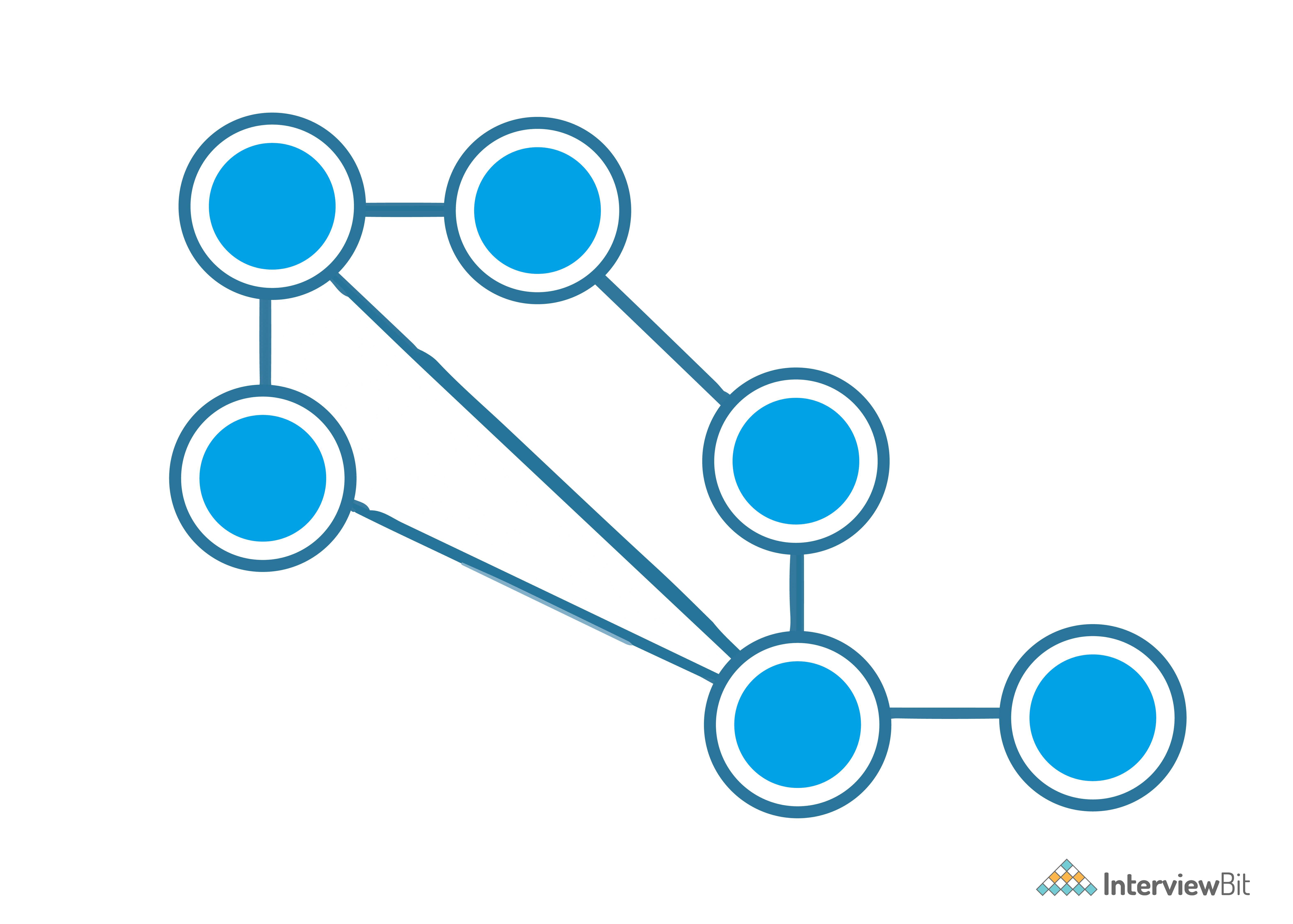
* All the nodes are connected to one single node known as the central node.
* It is more robust.
* If the central node fails the complete network is damaged.
* Easy to troubleshoot.
* Mainly used in home and office networks.

**Ring Topology:**

Ring Topology

* Each node is connected to exactly two nodes forming a ring structure
* If one of the nodes are damaged, it will damage the whole network
* It is used very rarely as it is expensive and hard to install and manage

**Mesh Topology:**

Mesh Topology

* Each node is connected to one or many nodes.
* It is robust as failure in one link only disconnects that node.
* It is rarely used and installation and management are difficult.

**Tree Topology:**

Tree Topology

* A combination of star and bus topology also know as an extended bus topology.
* All the smaller star networks are connected to a single bus.
* If the main bus fails, the whole network is damaged.

**Hybrid:**

* It is a combination of different topologies to form a new topology.
* It helps to ignore the drawback of a particular topology and helps to pick the strengths from other.

### 10. What is an IPv4 address? What are the different classes of IPv4?

An IP address is a 32-bit dynamic address of a node in the network. An IPv4 address has 4 octets of 8-bit each with each number with a value up to 255.

IPv4 classes are differentiated based on the number of hosts it supports on the network. There are five types of IPv4 classes and are based on the first octet of IP addresses which are classified as Class A, B, C, D, or E.

| **IPv4 Class** | **IPv4 Start Address** | **IPv4 End Address** | **Usage** |
| --- | --- | --- | --- |
| A | 0.0.0.0 | 127.255.255.255 | Used for Large Network |
| B | 128.0.0.0 | 191.255.255.255 | Used for Medium Size Network |
| C | 192.0.0.0 | 223.255.255.255 | Used for Local Area Network |
| D | 224.0.0.0 | 239.255.255.255 | Reserved for Multicasting |
| E | 240.0.0.0 | 255.255.255.254 | Study and R&D |

### 11. What are Private and Special IP addresses?

**Private Address:** For each class, there are specific IPs that are reserved specifically for private use only. This IP address cannot be used for devices on the Internet as they are non-routable.

| **IPv4 Class** | **Private IPv4 Start Address** | **Private IPv4 End Address** |
| --- | --- | --- |
| A | 10.0.0.0 | 10.255.255.255 |
| B | 172.16.0.0 | 172.31.255.255 |
| C | 192.168.0.0 | 192.168.255.255 |

**Special Address:** IP Range from 127.0.0.1 to 127.255.255.255 are network testing addresses also known as loopback addresses are the special IP address.

## Intermediate Interview Questions

### 12. Describe the OSI Reference Model

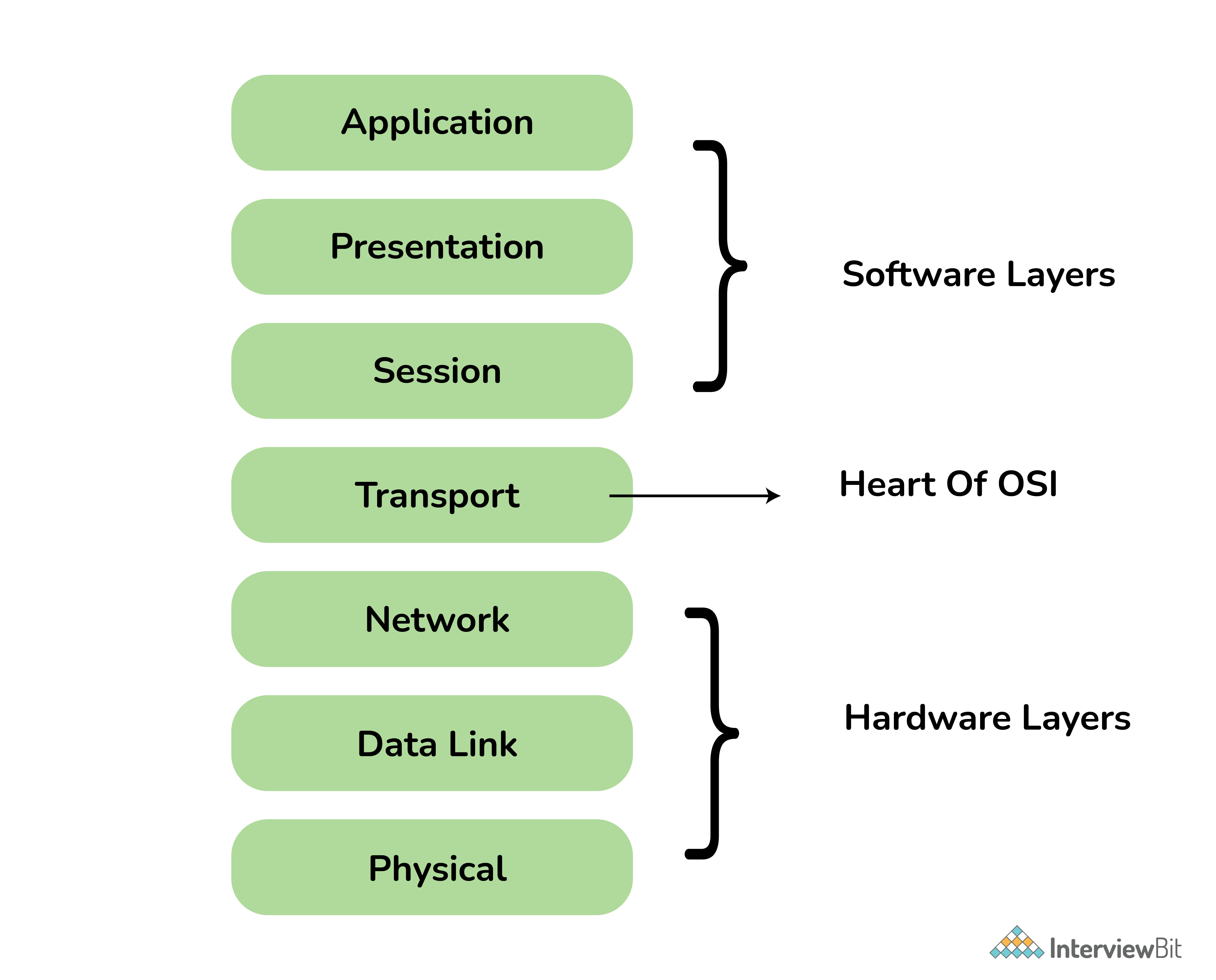
Open System Interconnections (OSI) is a network architecture model based on the ISO standards. It is called the OSI model as it deals with connecting the systems that are open for communication with other systems.

The OSI model has seven layers. The principles used to arrive at the seven layers can be summarized  briefly as below:

* Create a new layer if a different abstraction is needed.
* Each layer should have a well-defined function.
* The function of each layer is chosen based on internationally standardized protocols.

### 13. Define the 7 different layers of the OSI Reference Model

Here the 7 layers of the OSI reference model:

Layers of OSI Model

| **Layer** | **Unit Exchanged** | **Description** |
| --- | --- | --- |
| Physical | Bit | * It is concerned with transmitting raw bits over a communication channel. * Chooses which type of transmission mode is to be selected for the transmission. The available transmission modes are Simplex, Half Duplex and Full Duplex., |
| Data Link | Frame | * The main task of this layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors. * It also allows detecting damaged packets using the CRC (Cyclic Redundancy Check) error-detecting, code. * When more than one node is connected to a shared link, Data Link Layer protocols are required to determine which device has control over the link at a given time. * It is implemented by protocols like CSMA/CD, CSMA/CA, ALOHA, and Token Passing. |
| Network | Packet | * It controls the operation of the subnet. * The network layer takes care of feedback messaging through ICMP messages. |
| Transport | TPDU - Transaction Protocol Data Unit | * The basic functionality of this layer is to accept data from the above layers, split it up into smaller units if needed, pass these to the network layer, and ensure that all the pieces arrive correctly at the other end. * The Transport Layer takes care of Segmentation and Reassembly. |
| Session | SPDU - Session Protocol Data Unit | * The session layer allows users on different machines to establish sessions between them. * Dialogue control is using the full-duplex link as half-duplex. It sends out dummy packets from the client to the server when the client is ideal. |
| Presentation | PPDU - Presentation Protocol Data Unit | * The presentation layer is concerned with the syntax and semantics of the information transmitted. * It translates a message from a common form to the encoded format which will be understood by the receiver. |
| Application | APDU - Application Protocol Data Unit | * It contains a variety of protocols that are commonly needed by users. * The application layer sends data of any size to the transport layer. |

### 14. Describe the TCP/IP Reference Model

It is a compressed version of the OSI model with only 4 layers. It was developed by the US Department of Defence (DoD) in the 1980s. The name of this model is based on 2 standard protocols used i.e. TCP (Transmission Control Protocol) and IP (Internet Protocol).

### 15. Define the 4 different layers of the TCP/IP Reference Model

Layers of TCP/IP

| **Layer** | **Description** |
| --- | --- |
| Link | Decides which links such as serial lines or classic Ethernet must be used to meet the needs of the connectionless internet layer. |
| Internet | * The internet layer is the most important layer which holds the whole architecture together. * It delivers the IP packets where they are supposed to be delivered. |
| Transport | Its functionality is almost the same as the OSI transport layer. It enables peer entities on the network to carry on a conversation. |
| Application | It contains all the higher-level protocols. |

### 16. Differentiate OSI Reference Model with TCP/IP Reference Model

OSI Vs TCP/IP

| **OSI Reference Model** | **TCP/IP Reference Model** |
| --- | --- |
| 7 layered architecture | 4 layered architecture |
| Fixed boundaries and functionality for each layer | Flexible architecture with no strict boundaries between layers |
| Low Reliability | High Reliability |
| Vertical Layer Approach | Horizontal Layer Approach |

### 17. What are the HTTP and the HTTPS protocol?

HTTP is the HyperText Transfer Protocol which defines the set of rules and standards on how the information can be transmitted on the World Wide Web (WWW).  It helps the web browsers and web servers for communication. It is a ‘stateless protocol’ where each command is independent with respect to the previous command. HTTP is an application layer protocol built upon the TCP. It uses port 80 by default.

HTTPS is the HyperText Transfer Protocol Secure or Secure HTTP. It is an advanced and secured version of HTTP. On top of HTTP, SSL/TLS protocol is used to provide security. It enables secure transactions by encrypting the communication and also helps identify network servers securely. It uses port 443 by default.

### 18. What is the SMTP protocol?

SMTP is the Simple Mail Transfer Protocol. SMTP sets the rule for communication between servers. This set of rules helps the software to transmit emails over the internet. It supports both End-to-End and Store-and-Forward methods. It is in always-listening mode on port 25.

SMTP Protocol

### 19. What is the DNS?

DNS is the Domain Name System. It is considered as the devices/services directory of the Internet. It is a decentralized and hierarchical naming system for devices/services connected to the Internet. It translates the domain names to their corresponding IPs. For e.g. interviewbit.com to 172.217.166.36. It uses port 53 by default.

### 20. What is the use of a router and how is it different from a gateway?

The router is a networking device used for connecting two or more network segments. It directs the traffic in the network. It transfers information and data like web pages, emails, images, videos, etc. from source to destination in the form of packets. It operates at the network layer. The gateways are also used to route and regulate the network traffic but, they can also send data between two dissimilar networks while a router can only send data to similar networks.

## Advanced Interview Questions

### 21. What is the TCP protocol?

TCP or TCP/IP is the Transmission Control Protocol/Internet Protocol. It is a set of rules that decides how a computer connects to the Internet and how to transmit the data over the network. It creates a virtual network when more than one computer is connected to the network and uses the three ways handshake model to establish the connection which makes it more reliable.

### 22. What is the UDP protocol?

UDP is the User Datagram Protocol and is based on Datagrams. Mainly, it is used for multicasting and broadcasting. Its functionality is almost the same as TCP/IP Protocol except for the three ways of handshaking and error checking. It uses a simple transmission without any hand-shaking which makes it less reliable.

### 23. Compare between TCP and UDP

| **TCP/IP** | **UDP** |
| --- | --- |
| Connection-Oriented Protocol | Connectionless Protocol |
| More Reliable | Less Reliable |
| Slower Transmission | Faster Transmission |
| Packets order can be preserved or can be rearranged | Packets order is not fixed and packets are independent of each other |
| Uses three ways handshake model for connection | No handshake for establishing the connection |
| TCP packets are heavy-weight | UDP packets are light-weight |
| Offers error checking mechanism | No error checking mechanism |
| Protocols like HTTP, FTP, Telnet, SMTP, HTTPS, etc use TCP at the transport layer | Protocols like DNS, RIP, SNMP, RTP, BOOTP, TFTP, NIP, etc use UDP at the transport layer |

### 24. What is the ICMP protocol?

ICMP is the Internet Control Message Protocol. It is a network layer protocol used for error handling. It is mainly used by network devices like routers for diagnosing the network connection issues and crucial for error reporting and testing if the data is reaching the preferred destination in time. It uses port 7 by default.

### 25. What do you mean by the DHCP Protocol?

DHCP is the Dynamic Host Configuration Protocol.

It is an application layer protocol used to auto-configure devices on IP networks enabling them to use the TCP and UDP-based protocols. The DHCP servers auto-assign the IPs and other network configurations to the devices individually which enables them to communicate over the IP network. It helps to get the subnet mask, IP address and helps to resolve the DNS. It uses port 67 by default.

### 26. What is the ARP protocol?

ARP is Address Resolution Protocol. It is a network-level protocol used to convert the logical address i.e. IP address to the device's physical address i.e. MAC address. It can also be used to get the MAC address of devices when they are trying to communicate over the local network.

ARP Protocol

### 27. What is the FTP protocol?

FTP is a File Transfer Protocol. It is an application layer protocol used to transfer files and data reliably and efficiently between hosts. It can also be used to download files from remote servers to your computer. It uses port 27 by default.

### 28. What is the MAC address and how is it related to NIC?

MAC address is the Media Access Control address. It is a 48-bit or 64-bit unique identifier of devices in the network. It is also called the physical address embedded with Network Interface Card (NIC) used at the Data Link Layer. NIC is a hardware component in the networking device using which a device can connect to the network.

### 29. Differentiate the MAC address with the IP address

The difference between MAC address and IP address are as follows:

| **MAC Address** | **IP Address** |
| --- | --- |
| Media Access Control Address | Internet Protocol Address |
| 6 or 8-byte hexadecimal number | 4 (IPv4) or 16 (IPv6) Byte address |
| It is embedded with NIC | It is obtained from the network |
| Physical Address | Logical Address |
| Operates at Data Link Layer | Operates at Network Layer. |
| Helps to identify the device | Helps to identify the device connectivity on the network. |

### 30. What is a subnet?

A subnet is a network inside a network achieved by the process called subnetting which helps divide a network into subnets. It is used for getting a higher routing efficiency and enhances the security of the network. It reduces the time to extract the host address from the routing table.

### 31. Compare the hub vs switch

| **Hub** | **Switch** |
| --- | --- |
| Operates at Physical Layer | Operates at Data Link Layer |
| Half-Duplex transmission mode | Full-Duplex transmission mode |
| Ethernet devices can be connectedsend | LAN devices can be connected |
| Less complex, less intelligent, and cheaper | Intelligent and effective |
| No software support for the administration | Administration software support is present |
| Less speed up to 100 MBPS | Supports high speed in GBPS |
| Less efficient as there is no way to avoid collisions when more than one nodes sends the packets at the same time | More efficient as the collisions can be avoided or reduced as compared to Hub |

### 32. What is the difference between the ipconfig and the ifconfig?

| **ipconfig** | **ifconfig** |
| --- | --- |
| Internet Protocol Configuration | Interface Configuration |
| Command used in Microsoft operating systems to view and configure network interfaces | Command used in MAC, Linux, UNIX operating systems to view and configure network interfaces |
| Used to get the TCP/IP summary and allows to changes the DHCP and DNS settings | |

### 33. What is the firewall?

The firewall is a network security system that is used to monitor the incoming and outgoing traffic and blocks the same based on the firewall security policies. It acts as a wall between the internet (public network) and the networking devices (a private network). It is either a hardware device, software program, or a combination of both. It adds a layer of security to the network.

### 34. What are Unicasting, Anycasting, Multicasting and Broadcasting?

* **Unicasting:** If the message is sent to a single node from the source then it is known as unicasting. This is commonly used in networks to establish a new connection.
* **Anycasting:** If the message is sent to any of the nodes from the source then it is known as anycasting. It is mainly used to get the content from any of the servers in the Content Delivery System.
* **Multicasting:** If the message is sent to a subset of nodes from the source then it is known as multicasting. Used to send the same data to multiple receivers.
* **Broadcasting:** If the message is sent to all the nodes in a network from a source then it is known as broadcasting. DHCP and ARP in the local network use broadcasting.

### 35. What happens when you enter google.com in the web browser?

Below are the steps that are being followed:

* Check the browser cache first if the content is fresh and present in cache display the same.
* If not, the browser checks if the IP of the URL is present in the cache (browser and OS) if not then request the OS to do a DNS lookup using UDP to get the corresponding IP address of the URL from the DNS server to establish a new TCP connection.
* A new TCP connection is set between the browser and the server using three-way handshaking.
* An HTTP request is sent to the server using the TCP connection.
* The web servers running on the Servers handle the incoming HTTP request and send the HTTP response.
* The browser process the HTTP response sent by the server and may close the TCP connection or reuse the same for future requests.
* If the response data is cacheable then browsers cache the same.
* Browser decodes the response and renders the content.

36. **How are networks classified based on their connections?**

**Answer:** Networks are classified into two categories based on their connection types. **They are mentioned below:**

**Peer-to-peer networks (P2P):**When two or more computers are connected together to share resources without the use of a central server is termed as a peer-to-peer network. Computers in this type of network act as both server and client. It is generally used in small companies as they are not expensive.

**Server-based networks:**In this type of network, a central server is located to store the data, applications, etc of the clients. The server computer provides the security and network administration to the network.

**37. Define Pipelining?**

**Answer:** In Networking, when a task is in progress another task gets started before the previous task is finished. This is termed as Pipelining.

**38) What is an Encoder?**

**Answer:** Encoder is a circuit that uses an algorithm to convert any data or compress audio data or video data for transmission purposes. An encoder converts the analog signal into the digital signal.

**39) What is a Decoder?**

**Answer:** Decoder is a circuit that converts the encoded data to its actual format. It converts the digital signal into an analog signal.

**40) How can you recover the data from a system which is infected with a Virus?**

**Answer:** In another system (not infected with a virus) install an OS and antivirus with the latest updates. Then connect the HDD of the infected system as a secondary drive. Now scan the secondary HDD and clean it. Then copy the data into the system.

**41) Describe the key elements of the protocol?**

**Answer:** **Below are the 3 key elements of the protocol:**

**Syntax:**It is the format of the data. That means in which order the data is displayed.

**Semantics:**Describes the meaning of the bits in each section.

**Timing:**At what time the data is to be sent and how fast it is to be sent.

**42) Explain the difference between baseband and broadband transmission?**

**Answer:**

**Baseband Transmission:** A single signal consumes the whole bandwidth of the cable.

**Broadband Transmission:** Multiple signals of multiple frequencies are sent simultaneously.

**43) Expand SLIP?**

**Answer:**SLIP stands for Serial Line Interface Protocol. SLIP is a protocol used for transmitting IP datagrams over a serial line.

**44) What is SNMP?**

**Answer:** SNMP stands for Simple Network Management Protocol. It is a network protocol used for collecting organizing and exchanging information between network devices. SNMP is widely used in network management for configuring network devices like switches, hubs, routers, printers, servers.

**SNMP consists of the below components:**

* SNMP Manager
* Managed device
* SNMP Agent
* Management Information Base (MIB)

**45) What is NIC?**

**Answer:**NIC stands for Network Interface Card. It is also known as Network Adapter or Ethernet Card. It is in the form of an add-in card and is installed on a computer so that the computer can be connected to a network.

Each NIC has a MAC address which helps in identifying the computer on a network.

**C PROGRAMMING:**

1) What is C language?

C is a mid-level and procedural programming language. The Procedural programming language is also known as the structured programming language is a technique in which large programs are broken down into smaller modules, and each module uses structured code. This technique minimizes error and misinterpretation.

2) Why is C known as a mother language?

C is known as a mother language because most of the compilers and JVMs are written in C language. Most of the languages which are developed after C language has borrowed heavily from it like C++, Python, Rust, javascript, etc. It introduces new core concepts like arrays, functions, file handling which are used in these languages.

3) Why is C called a mid-level programming language?

C is called a mid-level programming language because it binds the low level and high -level programming language. We can use C language as a System programming to develop the operating system as well as an Application programming to generate menu driven customer driven billing system.

4) Who is the founder of C language?

Dennis Ritchie

5) When was C language developed?

C language was developed in 1972 at bell laboratories of AT&T.

6) What are the features of the C language?

The main features of C language are given below:

* **Simple:** C is a simple language because it follows the structured approach, i.e., a program is broken into parts
* **Portable:** C is highly portable means that once the program is written can be run on any machine with little or no modifications.
* **Mid Level:** C is a mid-level programming language as it combines the low- level language with the features of the high-level language.
* **Structured:** C is a structured language as the C program is broken into parts.
* **Fast Speed:** C language is very fast as it uses a powerful set of data types and operators.
* **Memory Management:** C provides an inbuilt memory function that saves the memory and improves the efficiency of our program.
* **Extensible:** C is an extensible language as it can adopt new features in the future.

7) What is the use of printf() and scanf() functions?

**printf():** The printf() function is used to print the integer, character, float and string values on to the screen.

Following are the format specifier:

* **%d**: It is a format specifier used to print an integer value.
* **%s**: It is a format specifier used to print a string.
* **%c**: It is a format specifier used to display a character value.
* **%f**: It is a format specifier used to display a floating point value.

**scanf()**: The scanf() function is used to take input from the user.

8) What is the difference between the local variable and global variable in C?

Following are the differences between a local variable and global variable:

|  |  |  |
| --- | --- | --- |
| **Basis for comparison** | **Local variable** | **Global variable** |
| Declaration | A variable which is declared inside function or block is known as a local variable. | A variable which is declared outside function or block is known as a global variable. |
| Scope | The scope of a variable is available within a function in which they are declared. | The scope of a variable is available throughout the program. |
| Access | Variables can be accessed only by those statements inside a function in which they are declared. | Any statement in the entire program can access variables. |
| Life | Life of a variable is created when the function block is entered and destroyed on its exit. | Life of a variable exists until the program is executing. |
| Storage | Variables are stored in a stack unless specified. | The compiler decides the storage location of a variable. |

9) What is the use of a static variable in C?

Following are the uses of a static variable:

* A variable which is declared as static is known as a static variable. The static variable retains its value between multiple function calls.
* Static variables are used because the scope of the static variable is available in the entire program. So, we can access a static variable anywhere in the program.
* The static variable is initially initialized to zero. If we update the value of a variable, then the updated value is assigned.
* The static variable is used as a common value which is shared by all the methods.
* The static variable is initialized only once in the memory heap to reduce the memory usage.

10) What is the use of the function in C?

**Uses of C function are:**

* C functions are used to avoid the rewriting the same code again and again in our program.
* C functions can be called any number of times from any place of our program.
* When a program is divided into functions, then any part of our program can easily be tracked.
* C functions provide the reusability concept, i.e., it breaks the big task into smaller tasks so that it makes the C program more understandable.

11) What is the difference between call by value and call by reference in C?

**Following are the differences between a call by value and call by reference are:**

|  |  |  |
| --- | --- | --- |
|  | **Call by value** | **Call by reference** |
| Description | When a copy of the value is passed to the function, then the original value is not modified. | When a copy of the value is passed to the function, then the original value is modified. |
| Memory location | Actual arguments and formal arguments are created in separate memory locations. | Actual arguments and formal arguments are created in the same memory location. |
| Safety | In this case, actual arguments remain safe as they cannot be modified. | In this case, actual arguments are not reliable, as they are modified. |
| Arguments | The copies of the actual arguments are passed to the formal arguments. | The addresses of actual arguments are passed to their respective formal arguments. |

**Example of call by value:**

1. #include <stdio.h>
2. **void** change(**int**,**int**);
3. **int** main()
4. {
5. **int** a=10,b=20;
6. change(a,b); //calling a function by passing the values of variables.
7. printf("Value of a is: %d",a);
8. printf("\n");
9. printf("Value of b is: %d",b);
10. **return** 0;
11. }
12. **void** change(**int** x,**int** y)
13. {
14. x=13;
15. y=17;
16. }

**Output:**

Value of a is: 10

Value of b is: 20

**Example of call by reference:**

1. #include <stdio.h>
2. **void** change(**int**\*,**int**\*);
3. **int** main()
4. {
5. **int** a=10,b=20;
6. change(&a,&b); // calling a function by passing references of variables.
7. printf("Value of a is: %d",a);
8. printf("\n");
9. printf("Value of b is: %d",b);
10. **return** 0;
11. }
12. **void** change(**int** \*x,**int** \*y)
13. {
14. \*x=13;
15. \*y=17;
16. }

**Output:**

Value of a is: 13

Value of b is: 17

12) What is recursion in C?

When a function calls itself, and this process is known as recursion. The function that calls itself is known as a recursive function.

Recursive function comes in two phases:

1. Winding phase
2. Unwinding phase

**Winding phase**: When the recursive function calls itself, and this phase ends when the condition is reached.

**Unwinding phase**: Unwinding phase starts when the condition is reached, and the control returns to the original call.

**Example of recursion**

1. #include <stdio.h>
2. **int** calculate\_fact(**int**);
3. **int** main()
4. {
5. **int** n=5,f;
6. f=calculate\_fact(n); // calling a function
7. printf("factorial of a number is %d",f);
8. **return** 0;
9. }
10. **int** calculate\_fact(**int** a)
11. {
12. **if**(a==1)
13. {
14. **return** 1;
15. }
16. **else**
17. **return** a\*calculate\_fact(a-1); //calling a function recursively.
18. }

**Output:**

factorial of a number is 120

13) What is an array in C?

An Array is a group of similar types of elements. It has a contiguous memory location. It makes the code optimized, easy to traverse and easy to sort. The size and type of arrays cannot be changed after its declaration.

**Arrays are of two types:**

* **One-dimensional array**: One-dimensional array is an array that stores the elements one after the another.

**Syntax:**

1. data\_type array\_name[size];

* **Multidimensional array**: Multidimensional array is an array that contains more than one array.

**Syntax:**

1. data\_type array\_name[size];

**Example of an array:**

1. #include <stdio.h>
2. **int** main()
3. {
4. **int** arr[5]={1,2,3,4,5}; //an array consists of five integer values.
5. **for**(**int** i=0;i<5;i++)
6. {
7. printf("%d ",arr[i]);
8. }
9. **return** 0;
10. }

**Output:**

1 2 3 4 5

14) What is a pointer in C?

A pointer is a variable that refers to the address of a value. It makes the code optimized and makes the performance fast. Whenever a variable is declared inside a program, then the system allocates some memory to a variable. The memory contains some address number. The variables that hold this address number is known as the pointer variable.

**For example:**

1. Data\_type \*p;

The above syntax tells that p is a pointer variable that holds the address number of a given data type value.

**Example of pointer**

1. #include <stdio.h>
2. **int** main()
3. {
4. **int** \*p; //pointer of type integer.
5. **int** a=5;
6. p=&a;
7. printf("Address value of 'a' variable is %u",p);
8. **return** 0;
9. }

**Output:**

Address value of 'a' variable is 428781252

15) What is the usage of the pointer in C?

* **Accessing array elements**: Pointers are used in traversing through an array of integers and strings. The string is an array of characters which is terminated by a null character '\0'.
* **Dynamic memory allocation**: Pointers are used in allocation and deallocation of memory during the execution of a program.
* **Call by Reference**: The pointers are used to pass a reference of a variable to other function.
* **Data Structures like a tree, graph, linked list, etc.**: The pointers are used to construct different data structures like tree, graph, linked list, etc.

16) What is a NULL pointer in C?

A pointer that doesn't refer to any address of value but NULL is known as a NULL pointer. When we assign a '0' value to a pointer of any type, then it becomes a Null pointer.

17) What is a far pointer in C?

A pointer which can access all the 16 segments (whole residence memory) of RAM is known as far pointer. A far pointer is a 32-bit pointer that obtains information outside the memory in a given section.

18) What is dangling pointer in C?

* If a pointer is pointing any memory location, but meanwhile another pointer deletes the memory occupied by the first pointer while the first pointer still points to that memory location, the first pointer will be known as a dangling pointer. This problem is known as a dangling pointer problem.
* Dangling pointer arises when an object is deleted without modifying the value of the pointer. The pointer points to the deallocated memory.

**Let's see this through an example.**

1. #include<stdio.h>
2. **void** main()
3. {
4. **int** \*ptr = malloc(constant value); //allocating a memory space.
5. free(ptr); //ptr becomes a dangling pointer.
6. }

In the above example, initially memory is allocated to the pointer variable ptr, and then the memory is deallocated from the pointer variable. Now, pointer variable, i.e., ptr becomes a dangling pointer.

**How to overcome the problem of a dangling pointer**

The problem of a dangling pointer can be overcome by assigning a NULL value to the dangling pointer. Let's understand this through an example:

1. #include<stdio.h>
2. **void** main()
3. {
4. **int** \*ptr = malloc(constant value); //allocating a memory space.
5. free(ptr); //ptr becomes a dangling pointer.
6. ptr=NULL; //Now, ptr is no longer a dangling pointer.
7. }

In the above example, after deallocating the memory from a pointer variable, ptr is assigned to a NULL value. This means that ptr does not point to any memory location. Therefore, it is no longer a dangling pointer.

19) What is pointer to pointer in C?

In case of a pointer to pointer concept, one pointer refers to the address of another pointer. The pointer to pointer is a chain of pointers. Generally, the pointer contains the address of a variable. The pointer to pointer contains the address of a first pointer. Let's understand this concept through an example:

1. #include <stdio.h>
2. **int** main()
3. {
4. **int** a=10;
5. **int** \*ptr,\*\*pptr; // \*ptr is a pointer and \*\*pptr is a double pointer.
6. ptr=&a;
7. pptr=&ptr;
8. printf("value of a is:%d",a);
9. printf("\n");
10. printf("value of \*ptr is : %d",\*ptr);
11. printf("\n");
12. printf("value of \*\*pptr is : %d",\*\*pptr);
13. **return** 0;
14. }

In the above example, pptr is a double pointer pointing to the address of the ptr variable and ptr points to the address of 'a' variable.

20) What is static memory allocation?

* In case of static memory allocation, memory is allocated at compile time, and memory can't be increased while executing the program. It is used in the array.
* The lifetime of a variable in static memory is the lifetime of a program.
* The static memory is allocated using static keyword.
* The static memory is implemented using stacks or heap.
* The pointer is required to access the variable present in the static memory.
* The static memory is faster than dynamic memory.
* In static memory, more memory space is required to store the variable.

1. For example:
2. **int** a[10];

The above example creates an array of integer type, and the size of an array is fixed, i.e., 10.

21) What is dynamic memory allocation?

* In case of dynamic memory allocation, memory is allocated at runtime and memory can be increased while executing the program. It is used in the linked list.
* The malloc() or calloc() function is required to allocate the memory at the runtime.
* An allocation or deallocation of memory is done at the execution time of a program.
* No dynamic pointers are required to access the memory.
* The dynamic memory is implemented using data segments.
* Less memory space is required to store the variable.

1. For example
2. **int** \*p= malloc(**sizeof**(**int**)\*10);

The above example allocates the memory at runtime.

22) What functions are used for dynamic memory allocation in C language?

1. malloc()
   * The malloc() function is used to allocate the memory during the execution of the program.
   * It does not initialize the memory but carries the garbage value.
   * It returns a null pointer if it could not be able to allocate the requested space.

**Syntax**

* 1. ptr = (cast-type\*) malloc(byte-size) // allocating the memory using malloc() function.

1. calloc()
   1. The calloc() is same as malloc() function, but the difference only is that it initializes the memory with zero value.

**Syntax**

* 1. ptr = (cast-type\*)calloc(n, element-size);// allocating the memory using calloc() function.

1. realloc()
   1. The realloc() function is used to reallocate the memory to the new size.
   2. If sufficient space is not available in the memory, then the new block is allocated to accommodate the existing data.

**Syntax**

* 1. ptr = realloc(ptr, newsize); // updating the memory size using realloc() function.

In the above syntax, ptr is allocated to a new size.

1. free():The free() function releases the memory allocated by either calloc() or malloc() function.

**Syntax**

* 1. free(ptr); // memory is released using free() function.

The above syntax releases the memory from a pointer variable ptr.

23) What is the difference between malloc() and calloc()?

|  |  |  |
| --- | --- | --- |
|  | **calloc()** | **malloc()** |
| Description | The malloc() function allocates a single block of requested memory. | The calloc() function allocates multiple blocks of requested memory. |
| Initialization | It initializes the content of the memory to zero. | It does not initialize the content of memory, so it carries the garbage value. |
| Number of arguments | It consists of two arguments. | It consists of only one argument. |
| Return value | It returns a pointer pointing to the allocated memory. | It returns a pointer pointing to the allocated memory. |

24) What is the structure?

* The structure is a user-defined data type that allows storing multiple types of data in a single unit. It occupies the sum of the memory of all members.
* The structure members can be accessed only through structure variables.
* Structure variables accessing the same structure but the memory allocated for each variable will be different.

**Syntax of structure**

1. **struct** structure\_name
2. {
3. Member\_variable1;
4. Member\_variable2
5. .
6. .
7. }[structure variables];

**Let's see a simple example.**

1. #include <stdio.h>
2. **struct** student
3. {
4. **char** name[10];       // structure members declaration.
5. **int** age;
6. }s1;      //structure variable
7. **int** main()
8. {
9. printf("Enter the name");
10. scanf("%s",s1.name);
11. printf("\n");
12. printf("Enter the age");
13. scanf("%d",&s1.age);
14. printf("\n");
15. printf("Name and age of a student: %s,%d",s1.name,s1.age);
16. **return** 0;
17. }

**Output:**

Enter the name shikha

Enter the age 26

Name and age of a student: shikha,26

25) What is a union?

* The union is a user-defined data type that allows storing multiple types of data in a single unit. However, it doesn't occupy the sum of the memory of all members. It holds the memory of the largest member only.
* In union, we can access only one variable at a time as it allocates one common space for all the members of a union.

**Syntax of union**

1. **union** union\_name
2. {
3. Member\_variable1;
4. Member\_variable2;
5. .
6. .
7. Member\_variable n;
8. }[**union** variables];

**Let's see a simple example**

1. #include<stdio.h>
2. union data
3. {
4. **int** a;      //union members declaration.
5. **float** b;
6. **char** ch;
7. };
8. **int** main()
9. {
10. union data d;       //union variable.
11. d.a=3;
12. d.b=5.6;
13. d.ch='a';
14. printf("value of a is %d",d.a);
15. printf("\n");
16. printf("value of b is %f",d.b);
17. printf("\n");
18. printf("value of ch is %c",d.ch);
19. **return** 0;
20. }

**Output:**

value of a is 1085485921

value of b is 5.600022

value of ch is a

In the above example, the value of a and b gets corrupted, and only variable ch shows the actual output. This is because all the members of a union share the common memory space. Hence, the variable ch whose value is currently updated.

26) What is an auto keyword in C?

In C, every local variable of a function is known as an automatic (auto) variable. Variables which are declared inside the function block are known as a local variable. The local variables are also known as an auto variable. It is optional to use an auto keyword before the data type of a variable. If no value is stored in the local variable, then it consists of a garbage value.

27) What is the purpose of sprintf() function?

The sprintf() stands for "string print." The sprintf() function does not print the output on the console screen. It transfers the data to the buffer. It returns the total number of characters present in the string.

**Syntax**

1. **int** sprintf ( **char** \* str, **const** **char** \* format, ... );

**Let's see a simple example**

1. #include<stdio.h>
2. **int** main()
3. {
4. **char** a[20];
5. **int** n=sprintf(a,"javaToint");
6. printf("value of n is %d",n);
7. **return** 0;}

**Output:**

value of n is 9

28) Can we compile a program without main() function?

Yes, we can compile, but it can't be executed.

But, if we use #define, we can compile and run a C program without using the main() function. For example:

1. #include<stdio.h>
2. #define start main
3. **void** start() {
4. printf("Hello");
5. }

29) What is a token?

The Token is an identifier. It can be constant, keyword, string literal, etc. A token is the smallest individual unit in a program. C has the following tokens:

1. Identifiers: Identifiers refer to the name of the variables.
2. Keywords: Keywords are the predefined words that are explained by the compiler.
3. Constants: Constants are the fixed values that cannot be changed during the execution of a program.
4. Operators: An operator is a symbol that performs the particular operation.
5. Special characters: All the characters except alphabets and digits are treated as special characters.

30) What is command line argument?

The argument passed to the main() function while executing the program is known as command line argument. For example:

1. main(**int** count, **char** \*args[]){
2. //code to  be executed
3. }

31) What is the acronym for ANSI?

The ANSI stands for " American National Standard Institute." It is an organization that maintains the broad range of disciplines including photographic film, computer languages, data encoding, mechanical parts, safety and more.

32) What is the difference between getch() and getche()?

The **getch()** function reads a single character from the keyboard. It doesn't use any buffer, so entered data will not be displayed on the output screen.

The **getche()** function reads a single character from the keyword, but data is displayed on the output screen. Press Alt+f5 to see the entered character.

**Let's see a simple example**

1. #include<stdio.h>
2. #include<conio.h>
3. **int** main()
4. {
6. **char** ch;
7. printf("Enter a character ");
8. ch=getch(); // taking an user input without printing the value.
9. printf("\nvalue of ch is %c",ch);
10. printf("\nEnter a character again ");
11. ch=getche(); // taking an user input and then displaying it on the screen.
12. printf("\nvalue of ch is %c",ch);
13. **return** 0;
14. }

**Output:**

Enter a character

value of ch is a

Enter a character again a

value of ch is a

In the above example, the value entered through a getch() function is not displayed on the screen while the value entered through a getche() function is displayed on the screen.

33) What is the newline escape sequence?

The new line escape sequence is represented by "\n". It inserts a new line on the output screen.

34) Who is the main contributor in designing the C language after Dennis Ritchie?

Brain Kernighan.

35) What is the difference between near, far and huge pointers?

A virtual address is composed of the *selector* and *offset*.

A **near** pointer doesn't have explicit selector whereas **far, and huge** pointers have explicit selector. When you perform pointer arithmetic on the far pointer, the selector is not modified, but in case of a huge pointer, it can be modified.

These are the non-standard keywords and implementation specific. These are irrelevant in a modern platform.

36) What is the maximum length of an identifier?

It is 32 characters ideally but implementation specific.

37) What is typecasting?

The typecasting is a process of converting one data type into another is known as typecasting. If we want to store the floating type value to an int type, then we will convert the data type into another data type explicitly.

**Syntax**

1. (type\_name) expression;

38) What are the functions to open and close the file in C language?

The *fopen()* function is used to open file whereas *fclose()* is used to close file.

39) Can we access the array using a pointer in C language?

Yes, by holding the base address of array into a pointer, we can access the array using a pointer.

40) What is an infinite loop?

A loop running continuously for an indefinite number of times is called the infinite loop.

**Infinite For Loop:**

1. **for**(;;){
2. //code to be executed
3. }

**Infinite While Loop:**

1. **while**(1){
2. //code to be executed
3. }

**Infinite Do-While Loop:**

1. **do**{
2. //code to be executed
3. }**while**(1);

41) Write a program to print "hello world" without using a semicolon?

1. #include<stdio.h>
2. **void** main(){
3. **if**(printf("hello world")){} // It prints the ?hello world? on the screen.
4. }

[More details.](https://www.javatpoint.com/c-program-to-print-hello-without-semicolon)

42) Write a program to swap two numbers without using the third variable?

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5. **int** a=10, b=20;    //declaration of variables.
6. clrscr();        //It clears the screen.
7. printf("Before swap a=%d b=%d",a,b);
9. a=a+b;//a=30 (10+20)
10. b=a-b;//b=10 (30-20)
11. a=a-b;//a=20 (30-10)
13. printf("\nAfter swap a=%d b=%d",a,b);
14. getch();
15. }

[More details.](https://www.javatpoint.com/c-program-to-swap-two-numbers-without-using-third-variable)

43) Write a program to print Fibonacci series without using recursion?

1. #include<stdio.h>
2. #include<conio.h>
3. **void** main()
4. {
5. **int** n1=0,n2=1,n3,i,number;
6. clrscr();
7. printf("Enter the number of elements:");
8. scanf("%d",&number);
9. printf("\n%d %d",n1,n2);//printing 0 and 1
11. **for**(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already printed
12. {
13. n3=n1+n2;
14. printf(" %d",n3);
15. n1=n2;
16. n2=n3;
17. }
18. getch();
19. }

44) Write a program to print Fibonacci series using recursion?

1. #include<stdio.h>
2. #include<conio.h>
3. **void** printFibonacci(**int** n) // function to calculate the fibonacci series of a given number.
4. {
5. **static** **int** n1=0,n2=1,n3;    // declaration of static variables.
6. **if**(n>0){
7. n3 = n1 + n2;
8. n1 = n2;
9. n2 = n3;
10. printf("%d ",n3);
11. printFibonacci(n-1);    //calling the function recursively.
12. }
13. }
14. **void** main(){
15. **int** n;
16. clrscr();
17. printf("Enter the number of elements: ");
18. scanf("%d",&n);
19. printf("Fibonacci Series: ");
20. printf("%d %d ",0,1);
21. printFibonacci(n-2);//n-2 because 2 numbers are already printed
22. getch();
23. }

45) Write a program to check prime number in C Programming?

1. #include<stdio.h>
2. #include<conio.h>
3. **void** main()
4. {
5. **int** n,i,m=0,flag=0;    //declaration of variables.
6. clrscr();    //It clears the screen.
7. printf("Enter the number to check prime:");
8. scanf("%d",&n);
9. m=n/2;
10. **for**(i=2;i<=m;i++)
11. {
12. **if**(n%i==0)
13. {
14. printf("Number is not prime");
15. flag=1;
16. **break**;    //break keyword used to terminate from the loop.
17. }
18. }
19. **if**(flag==0)
20. printf("Number is prime");
21. getch();    //It reads a character from the keyword.
22. }

46) Write a program to check palindrome number in C Programming?

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5. **int** n,r,sum=0,temp;
6. clrscr();
7. printf("enter the number=");
8. scanf("%d",&n);
9. temp=n;
10. **while**(n>0)
11. {
12. r=n%10;
13. sum=(sum\*10)+r;
14. n=n/10;
15. }
16. **if**(temp==sum)
17. printf("palindrome number ");
18. **else**
19. printf("not palindrome");
20. getch();
21. }

47) Write a program to print factorial of given number without using recursion?

1. #include<stdio.h>
2. #include<conio.h>
3. **void** main(){
4. **int** i,fact=1,number;
5. clrscr();
6. printf("Enter a number: ");
7. scanf("%d",&number);
9. **for**(i=1;i<=number;i++){
10. fact=fact\*i;
11. }
12. printf("Factorial of %d is: %d",number,fact);
13. getch();
14. }

48) Write a program to print factorial of given number using recursion?

1. #include<stdio.h>
2. #include<conio.h>
3. **long** factorial(**int** n)    // function to calculate the factorial of a given number.
4. {
5. **if** (n == 0)
6. **return** 1;
7. **else**
8. **return**(n \* factorial(n-1));    //calling the function recursively.
9. }
10. **void** main()
11. {
12. **int** number;    //declaration of variables.
13. **long** fact;
14. clrscr();
15. printf("Enter a number: ");
16. scanf("%d", &number);
17. fact = factorial(number);    //calling a function.
18. printf("Factorial of %d is %ld\n", number, fact);
19. getch();   //It reads a character from the keyword.
20. }

49) Write a program to check Armstrong number in C?

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5. **int** n,r,sum=0,temp;    //declaration of variables.
6. clrscr(); //It clears the screen.
7. printf("enter the number=");
8. scanf("%d",&n);
9. temp=n;
10. **while**(n>0)
11. {
12. r=n%10;
13. sum=sum+(r\*r\*r);
14. n=n/10;
15. }
16. **if**(temp==sum)
17. printf("armstrong  number ");
18. **else**
19. printf("not armstrong number");
20. getch();  //It reads a character from the keyword.
21. }

50) Write a program to reverse a given number in C?

1. #include<stdio.h>
2. #include<conio.h>
3. main()
4. {
5. **int** n, reverse=0, rem;    //declaration of variables.
6. clrscr(); // It clears the screen.
7. printf("Enter a number: ");
8. scanf("%d", &n);
9. **while**(n!=0)
10. {
11. rem=n%10;
12. reverse=reverse\*10+rem;
13. n/=10;
14. }
15. printf("Reversed Number: %d",reverse);
16. getch();  // It reads a character from the keyword.
17. }

**DATA STRUCTURES:**

### 1. Explain what is data structure?

The data structure is nothing but an entity where the data is perfectly aligned and can be manipulated as per the requirement. When we deal with data structure it is not just about one table of data but it is about different data sets and how well they are aligned with each other. Overall, it helps the data to be organized.

### 2. What are the basic operations performed on various data structures?

The basic operations performed on data structures are as follows:

* **Insertion** - Adds a new data element in the data structure.
* **Deletion** - Removes a data element in a data structure.
* **Searching** - Involves searching for a specified data element in a data structure.
* **Traversal** - Processing all data elements present in it.
* **Merging** - Combines two similar data structures to form a new data structure of the same type.
* **Sorting** - Arranging data elements of a data structure in a specified order.

### 3. Explain what is a linked list?

A linked list is nothing but a sequence of nodes. With this sequence, each node is connected to the following node. It forms a chain of data storage.

### 4. Explain the process of how do you reference all the elements in the one-dimension array in detail?

To reference all the elements in the one-dimension array, we have to use an indexed loop. With the help of this, it executes from “0” to array size minus one. By following this process the loop counter will be able to refer to all the elements.

### 5. List out the areas where the data structure is applied?

The data structure is a vital aspect while handling data. The following are specific areas where the data structure is applied:

* Numerical analysis
* Operating systems
* A.I.
* Database management
* Statistical analysis

The above are few areas where the data structure is applied and not limited to.

### 6. What is infix, prefix, and postfix in data structure?

The way to write arithmetic expressions is known as notation. There are three types of notations used in an arithmetic expression, i.e., without changing the essence or output of expression. These notations are:

* **Prefix (Polish) Notation** - In this, the operator is prefixed to operands, i.e. ahead of operands.
* **Infix Notation** - In this, operators are used in between operands.
* **Postfix (Reverse-Polish) Notation** - In this, the operator is postfixed to the operands, i.e., after the operands.

The following table briefly tries to show the difference in all three notations −

|  |  |  |
| --- | --- | --- |
| **Infix Notation** | **Prefix Notation** | **Postfix Notation** |
| x + y | + x y | x y + |
| (x + y) ∗ z | ∗ + x y z | x y + z ∗ |
| x ∗ (y + z) | ∗ x + y z | x y z + ∗ |
| x / y + z / w | + / x y / z w | x y / z w / + |
| (x + y) ∗ (z + w) | ∗ + x y + z w | x y + z w + ∗ |
| ((x + y) ∗ z) - w | - ∗ + x y z w | x y + z ∗ w - |

## Data Structures Interview Questions and Answers for Freshers

### 7. Explain the terminology LIFO?

LIFO stands for **Last In First Out**.

This process describes how the data is accessed, stored, and then retrieved. So the latest data that is stored in the database can be extracted first.

### 8. Explain what is a binary tree?

 It is one type of data structure that has two nodes, has left node and a right node. In a programming language, binary trees are considered to be an extension to the linked list.

### 9. Define what is a stack?

The stack is considered as a data structure where the top layer element can be accessed. The data is stored in the stack and every time when data is stored, it pushes the data downwards which enables the users to access the latest data from the top layers.

### 10. Explain what is multidimensional arrays?

Multidimensional arrays use multiple indexes in order to store data in the database. In a few scenarios, data cannot be stored using a single dimension index, in these scenarios multidimensional arrays are useful.

### 11. Explain whether a linked list is considered as a linear or non-linear data structure?

This is purely determined on the requirement basis, a linked list can be considered as a linear data structure or a non-linear data structure. For example: If the linked list is used on storage, then the linked list is considered as a nonlinear data structure.

If linked lists are used against access strategies then they are considered as a linear data structure.

### 12. Explain how does dynamic memory allocation will help you in managing data?

A dynamic memory allocation will help you effectively manage your data by allocating structured blocks to have composite structures that can be flexible, i.e. it can expand and can contract based on the need.

Also, they are capable of storing simple structured data types.

### 13. What is FIFO?

FIFO in data terminology stands as “**First in, First Out**”.

This process defines or depicts how the data is stored inserted and accessed in a queue. Within this process, the data that is inserted at the beginning of the queue will only be extracted or accessed first.

### 14. Explain what is merge sort and how it is useful?

A merge sort is nothing but a process where the data is divided and sorted to reach the end goal. Within this process, the adjacent elements are merged and sorted to create bigger elements. These sorted elements are gathered again and made the even bigger list. This process is continuous and repetitive until and unless they have nailed it down to a single sorted list.

### 15. List out all the advantages of a linked list?

The important aspect or advantage of a linked list is that it is the perfect data structure where the data can be modified very easily. Also, it doesn’t matter how many elements are available on the linked list.

### 16. Explain the main difference between PUSH and a POP?

The two main activities, i.e. Pushing and Popping applies the way how data is stored and retrieved in an entity. So if you check in detail, a Push is nothing but a process where data is added to the stack.  On the contrary, a Pop is an activity where data is retrieved from the stack. When we discuss data retrieval it only considers the topmost available data.

### 17. Can you explain with an example how does a variable declaration activity will consume or affect the memory allocation?

The amount of space or memory is occupied or allocated depends upon the data type of the variables that are declared. So let’s explain the same by considering an example:  Let’s say the variable is declared as an integer type then 32 bits of memory storage is allocated for that particular variable.

So based on the data type of the variable, the memory space will be allocated.

|  |
| --- |
|  |

### 18. Define the advantages and disadvantages of the heap compared to a stack?

The advantages of the heap compared to a stack are listed below:

1. Heap is more flexible when compared to a stack
2. Memory space of the heap can actually be allocated and de-allocated as per the need.

On the contrary, the disadvantages of the heap compared to a stack is listed below:

1. The memory of the heap is slower when compared to the memory of the stack

### 19. Explain how new data can be inserted into the tree?

The following are the steps that you need to follow to insert the data into the tree:

1. First of all, check whether the data is unique or not ( i.e. check whether the data that you are going to insert doesn’t already exist in the tree).
2. Then check if the tree is empty. If the tree is empty then all you need to do is just insert a new item into the root.  If the key is smaller than that of a root’s key then insert that data into the root’s left subtree or otherwise, insert the data into the right side of the root’s subtree.

### 20. Can you tell me the minimum number of nodes that a binary tree can have?

A binary tree is allowed or can have a minimum of zero nodes. Further, a binary tree can also have 1 or 2 nodes.

### 21. Explain a little bit about the dynamic data structure?

The nature of the dynamic data structure is different compared to the standard data structures, the word dynamic data structures means that the data structure is flexible in nature. As per the need, the data structure can be expanded and contracted. Thus it helps the users to manipulate the data without worrying too much about the data structure flexibility.

### 22. Define what is an array?

While referring to array the data is stored and utilized based on the index and this number actually co-relates to the element number in the data sequence. So thus making it flexible to access data in any order. Within programming language, an array is considered as a variable having a certain number of indexed elements.

### 23. Can you tell me the minimum number of queues that are needed to implement a priority queue?

The minimum number of queues that are needed is two. Out of which, one queue is intended for sorting priorities and the other queue is meant for the actual storage of data.

### 24. List out all different sorting algorithms that are available and state which sorting algorithm is considered as the fastest?

The list of all sorting algorithms are below:

1. Quicksort
2. Bubble sort
3. Balloon sort
4. Radix sort
5. Merge sort

Out of the above sorting options, none of the sorting algorithms can be tagged as the fastest algorithm, because each of these sorting algorithms is defined for a specific purpose. So based on the data structure and data sets available the sorting algorithms are used.

### 25. Explain what is a dequeue?

A de queue is nothing but a double-ended queue. Within this structure, the elements can be inserted or deleted from both sides.

### 26. Explain the process of how a selection sort works?

A selection sort is a process where it picks up the smallest number from the entire data setlist and places it at the beginning. The same process is continued where the second position is already filled in. The same process is continued all the way until the list is completed. The selection sort is defined as a simple sort algorithm when compared to others.

## Data Structures Interview Questions and Answers for Experienced

### 27. Explain what is a graph?

A graph is nothing but a type of data structure that has a set of ordered pairs. In turn, these pairs are again acknowledged as edges or arcs. These are used to connect different nodes where the data can be accessed and stored based on the needs.

### 28. Is it possible to implement a stack using a queue?

Yes, you can implement a stack using two queues. Any data structure to act like a stack should have a push() method to add data on top and a pop() method to remove the top data.

### 29. How would you implement a queue using a stack?

Using two stacks, you can implement a queue. The purpose is to complete the queue's en queue operation so that the initially entered element always ends up at the top of the stack.

* First, to enqueue an item into the queue, migrate all the elements from the beginning stack to the second stack, push the item into the stack, and send all elements back to the first stack.
* To dequeue an item from the queue, return the top item from the first stack.

### 30. Where is the LRU cache used in data structure?

In data structures, you use LRU (Least Recently Used) cache to organize items in order of use, enabling you to quickly find out which item hasn't been used for a long time.

### 31. Which Data Structure is used to implement LRU cache?

To implement the LRU cache, you should use two data structures: a hashmap and a doubly linked list.

A hashmap helps with the lookup of cached keys, and a doubly-linked list helps maintain the eviction order.

### 32. What is the difference between an array and a linked list?

Array and Linked list are two ways of organizing the data in memory. The below table lists the various differences between the array and linked lists:

|  |  |
| --- | --- |
| **Array** | **Linked List** |
| An array is a sequence of elements of a similar data type. | A Linked list is a set of objects known as a node, where it internally consists of two parts, i.e., data and address. |
| It can be accessed irregularly using the array index. | Linked lists support random access. Only supports sequential access. |
| Array elements store in contiguous locations in memory. | New elements can be stored anywhere, and a reference is created for the new element using pointers. |
| In arrays, memory allocation is done during compile time. | In linked lists, memory allocation is done during runtime. |
| Array size must be defined at the time of declaration/initialization. | Linked list size grows when new elements are inserted or deleted. |

### 33. What are the advantages of Linked Lists over Array?

The following are the advantages of Linked Lists over Arrays:

* **Dynamic Data Structure** - A Linked list is a dynamic data structure so that it can grow at runtime by deallocating and allocating memory. There is no need to present the linked list initial size.
* **Insertion and Deletion Operations** - Insertion and Deletion operations are easier in linked lists. You need to have to update the address present in the next pointer of a node.
* **Implementation**- Data structures like queue, tree, and stack are easily implemented using the Linked list.
* **No memory wastage** - Efficient memory utilization can be achieved in the linked lists.

### 34. What are the applications of a stack in a data structure?

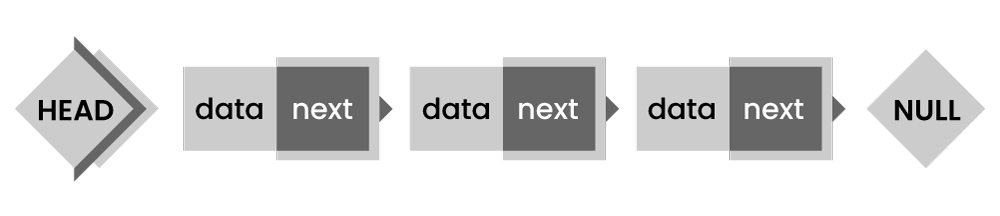
Following are some of the essential applications in a data structure:

* Expression evaluation
* Backtracking
* Function calling and return
* Memory management
* Checking parenthesis matching in an expression

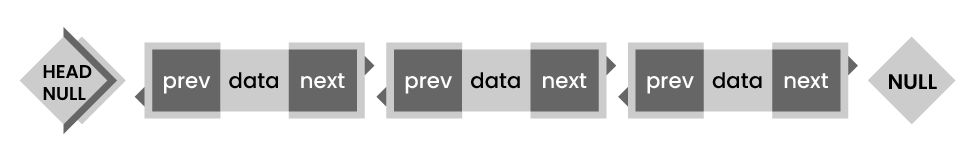
### 35. What are the different types of Linked List?

The following are the different types of Linked Lists.

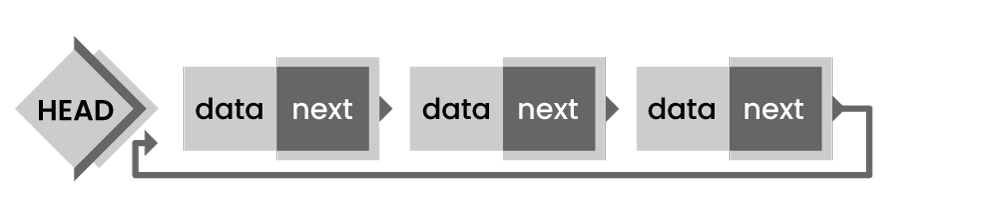
* **Singly Linked List**- This is the most common type, and each node has data and a pointer to the next node.



* **Doubly Linked List** - In this type, the pointer is added to the previous node, either in a forward or backward direction.



* **Circular Linked List** - In this type, the last element is linked to the first element.



### 36. What is the difference between storage structure and file structure?

The main difference between storage structure and file structure depends on the memory area that is accessed.

* **Storage structure:** It's a data structure representation in computer memory.
* **File structure:** It's a storage structure representation in the auxiliary memory.

### 37. What operations can be performed on a stack?

Mainly the following operations are performed on a stack:

* **Push operation:** To add an item to the stack. If the stack is complete, then it is in an overflow condition.
* **Pop operation:**It is used to remove an item from the stack. If it's an empty stack, then it is in underflow condition.
* **isEmpty operation:** If the stack is empty returns true, else false.
* **Peek or Top operation:** This returns the top element of the stack.

### 38. Name some applications of data structures?

* Operating system
* Artificial Intelligence
* Statistical analysis
* Database management
* Compiler design
* Graphics
* Simulation
* Numerical analysis

### 39. Explain about Linked List Data Structure.

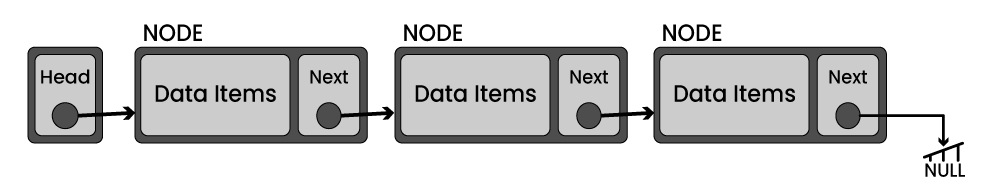
A linked list is a series of data structures connected via links. In simple words, it's a sequence of links that contain items. After the array, the linked list is the second most used data structure. The essential terms to understand the linked list are:

**Link** - In a linked list, each link stores data called an element.

**Next**- In a linked list, each link is connected to the following link called next.

**LinkedList**-  It contains the connection link to the first link called first.

The below diagram depicts how nodes are connected in the Linked List:



Basic operations supported by a linked list:

* **Insertion**- Inserts an element at the list beginning.
* **Deletion**- Deletes an element at the list beginning.
* **Display**- Displays the complete list.
* **Search**- Searches an element using the given key.
* **Delete**- Deletes an element using the given key.

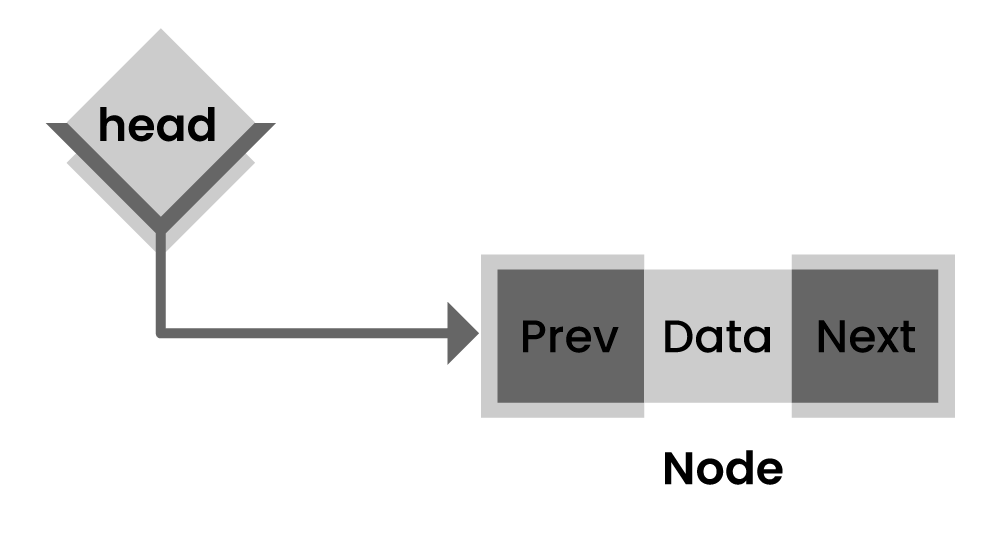
### 40. Are linked lists considered non-linear or linear data structures?

It depends on where you plan to use Linked lists. You can consider a linked list for both non-linear and linear data structures. When used for data storage, it is regarded as a non-linear data structure. When used for access strategies, it is considered a linear data structure.

### 41. What is a doubly-linked list used for?

A doubly linked list is one of the complex types of the linked list, where a node contains a pointer to the previous and the next node in the sequence. It consists of three parts: node data, a pointer to the next node in sequence (next pointer), a pointer to the previous node (previous pointer)

The below diagram depicts the working of doubly linked list:



Some of the real-time applications where doubly-linked lists used are navigation systems and browsers (when both back and front navigation is required).

|  |
| --- |
|  |

### 42. What is a queue in data structure?

A queue is a linear data structure that supports a specific order in which operations are performed. The order is FIFO (First in First Out) methodology, i.e., data items stored first will be accessed first. Unlike stack, the queue is open at both ends, and one end is always used to insert data and another one to remove data.

The basic operations associated with queues -

* **Dequeue** - To remove an item
* **Enqueue** - To insert an item
* **isempty()** − Confirms whether the queue is empty.
* **isfull()** − Confirms whether the queue is full.
* **peek()** − Gets the element at the front of the queue without removing it.

### 43. List a few queue data structure applications.

As the name suggests, the queue is used whenever you need to manage a group of objects in the order FIFO. A few of the queue data structure applications are listed below:

* Serving requests on a single shared resource, like CPU task scheduling, printer, etc.
* Handling interruptions in real-time systems.
* Buffers in apps like CD player and MP3 media players
* In maintaining a playlist in media players, like adding or removing songs.

### 44. What is the difference between stack and heap?

Both stack and heap are used for memory needs. The stack is primarily used to save the method execution order and local variables, and always follow the LIFO order.

Whereas heap is used for dynamic allocation and deallocation of memory blocks. It stores objects in Java. Memory allocated to the heap lives until one of the following events occurs:

* Memory free
* Program terminated

The size of heap memory is more when using recursion when compared with the stack, as it quickly fill-ups stack memory.

### 45. Name a few graph data structure applications.

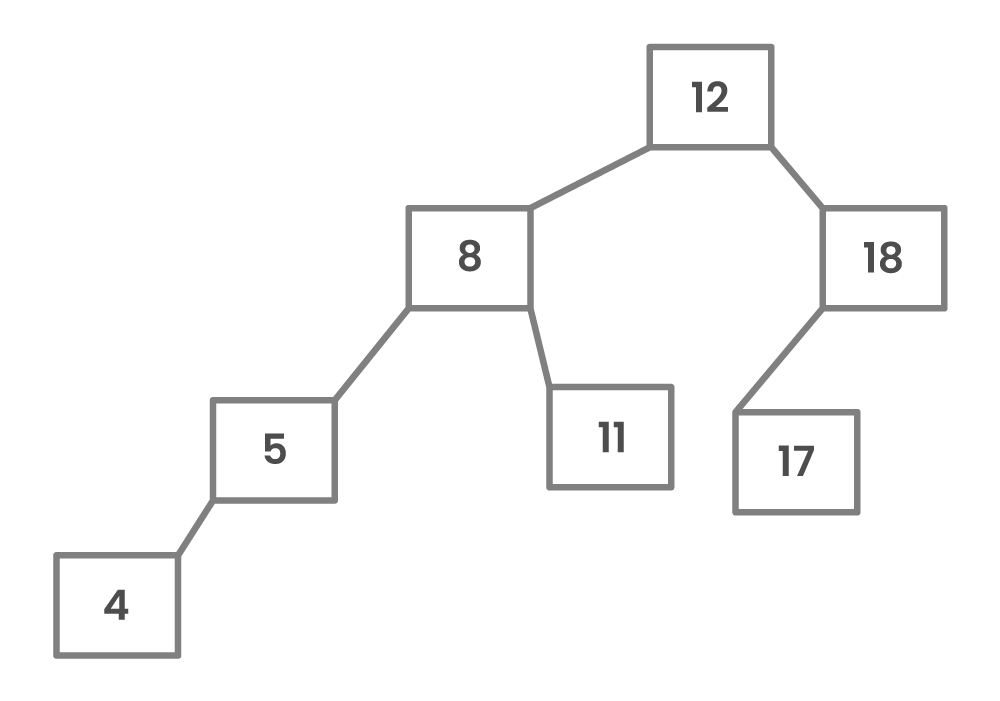
Applications of graph data structures in real-time are:

* Social graphs
* Path optimization algorithms
* Recommendation engines
* Scientific computations

### 46. What is an AVL tree?

An AVL (Adelson, Velskii, and Landi) is a self-balancing binary search tree where the variation of heights of the right and left subtrees of any node is not more than one.

An example of an AVL tree:



A tree is balanced if the balance factor of each node is between -1 to 1. Or else, the tree is unbalanced and needs to be balanced.

### 47. How do you detect a loop in a linked list?

* Using Floyd's cycle-finding Algorithm
* Using hashing
* Using the visited nodes method

### 48. What is a Jagged array?

Jagged arrays are a particular type of arrays used for storing rows of data of varying lengths to improve efficiency when working with multidimensional arrays.

### 49. What is the max heap in the data structure?

A max heap in a data structure is a complete binary tree where each internal node's value is greater than or equal to that node's children's values.

### 50. How to find the height of a node in a tree?

You can find the height of a binary tree using a recursive Depth-First Search (DFS) algorithm, as shown below:

* **Base case:** If there is no node, return 0.
* **Else:** If there are 1 or 2 children, return the peak of the height of the left and right subtrees, plus 1 to account for the current node.

### 51. List the types of trees.

* General Tree
* Binary Tree
* Forests
* Expression Tree
* Binary Search Tree
* Tournament Tree

### 52. Which data structures do you use in DFS and BFS algorithms?

* In the DFS algorithm, you use the Stack data structure.
* In the BFS algorithm, you use the Queue data structure.

OPERATING SYSTEMS:

**1) Explain the main purpose of an operating system?**

Operating systems exist for two main purposes. One is that it is designed to make sure a computer system performs well by managing its computational activities. Another is that it provides an environment for the development and execution of programs.

**2) What is demand paging?**

Demand paging is referred when not all of a process’s pages are in the RAM, then the OS brings the missing(and required) pages from the disk into the RAM.

**3) What are the advantages of a multiprocessor system?**

With an increased number of processors, there is a considerable increase in throughput. It can also save more money because they can share resources. Finally, overall reliability is increased as well.

**4) What is kernel?**

A kernel is the core of every operating system. It connects applications to the actual processing of data. It also manages all communications between software and hardware components to ensure usability and reliability.

**5) What are real-time systems?**

Real-time systems are used when rigid time requirements have been placed on the operation of a processor. It has well defined and fixed time constraints.

**6) What is a virtual memory?**

Virtual memory is a memory management technique for letting processes execute outside of memory. This is very useful especially is an executing program cannot fit in the physical memory.

**7) Describe the objective of multiprogramming.**

The main objective of multiprogramming is to have a process running at all times. With this design, CPU utilization is said to be maximized.

**8 ) What is time- sharing system?**

In a Time-sharing system, the CPU executes multiple jobs by switching among them, also known as multitasking. This process happens so fast that users can interact with each program while it is running.

**9) What is SMP?**

SMP is a short form of Symmetric Multi-Processing. It is the most common type of multiple-processor systems. In this system, each processor runs an identical copy of the operating system, and these copies communicate with one another as needed.

**10) How are server systems classified?**

Server systems can be classified as either computer-server systems or file server systems. In the first case, an interface is made available for clients to send requests to perform an action. In the second case, provisions are available for clients to create, access and update files.

**11) What is asymmetric clustering?**

In asymmetric clustering, a machine is in a state known as hot standby mode where it does nothing but to monitor the active server. That machine takes the active server’s role should the server fails.

**12) What is a thread?**

A thread is a basic unit of CPU utilization. In general, a thread is composed of a thread ID, program counter, register set, and the stack.

**13) Give some benefits of multithreaded programming.**

– there is increased responsiveness to the user  
– resource sharing within the process  
– economy  
– utilization of multiprocessing architecture

**14) Briefly explain FCFS.**

FCFS stands for First-come, first-served. It is one type of scheduling algorithm. In this scheme, the process that requests the CPU first is allocated the CPU first. Implementation is managed by a FIFO queue.

**15) What is RR scheduling algorithm?**

RR (round-robin) scheduling algorithm is primarily aimed for time-sharing systems. A circular queue is a setup in such a way that the CPU scheduler goes around that queue, allocating CPU to each process for a time interval of up to around 10 to 100 milliseconds.

**16) What are necessary conditions which can lead to a deadlock situation in a system?**

Deadlock situations occur when four conditions occur simultaneously in a system: Mutual exclusion; Hold and Wait; No preemption; and Circular wait.

**17) Enumerate the different RAID levels.**

RAID 0 – Non-redundant striping  
RAID 1 – Mirrored Disks  
RAID 2 – Memory-style error-correcting codes  
RAID 3 – Bit-interleaved Parity  
RAID 4 – Block-interleaved Parity  
RAID 5 – Block-interleaved distributed Parity  
RAID 6 – P+Q Redundancy

**18) Describe Banker’s algorithm**

Bankers Algorithm

Banker’s algorithm is one form of deadlock-avoidance in a system. It gets its name from a [banking](https://career.guru99.com/top-50-banking-interview-questions/) system wherein the bank never allocates available cash in such a way that it can no longer satisfy the needs of all of its customers.

**19) What factors determine whether a detection-algorithm must be utilized in a deadlock avoidance system?**

One is that it depends on how often a deadlock is likely to occur under the implementation of this algorithm. The other has to do with how many processes will be affected by deadlock when this algorithm is applied.

**20) State the main difference between logical from physical address space.**

Logical address refers to the address that is generated by the CPU. On the other hand, physical address refers to the address that is seen by the memory unit.

**21) How does dynamic loading aid in better memory space utilization?**

With dynamic loading, a routine is not loaded until it is called. This method is especially useful when large amounts of code are needed in order to handle infrequently occurring cases such as error routines.

**22) What are overlays?**

Overlays are used to enable a process to be larger than the amount of memory allocated to it. The basic idea of this is that only instructions and data that are needed at any given time are kept in memory.

**23) What is the basic function of paging?**

Paging is a memory management scheme that permits the physical address space of a process to be noncontiguous. It avoids the considerable problem of having to fit varied sized memory chunks onto the backing store.

**24) What is fragmentation?**

Fragmentation is memory wasted. It can be internal if we are dealing with systems that have fixed-sized allocation units, or external if we are dealing with systems that have variable-sized allocation units.

**25) How does swapping result in better memory management?**

During regular intervals that are set by the operating system, processes can be copied from main memory to a backing store, and then copied back later. Swapping allows more operations to be run that can fit into memory at one time.

**26) Give an example of a Process State.**

– New State – means a process is being created  
– Running – means instructions are being executed  
– Waiting – means a process is waiting for certain conditions or events to occur  
– Ready – means a process is waiting for an instruction from the main processor  
– Terminate – means a process is stopped abruptly

**27) What is a socket?**

A socket provides a connection between two applications. Each endpoint of a communication is a socket.

**28) What is Direct Access Method?**

Direct Access method is based on a disk model of a file, such that it is viewed as a numbered sequence of blocks or records. It allows arbitrary blocks to be read or written. Direct access is advantageous when accessing large amounts of information.

**29) When does thrashing occur?**

Thrashing refers to an instance of high paging activity. This happens when it is spending more time paging instead of executing.

**30) What is the best page size when designing an operating system?**

The best paging size varies from system to system, so there is no single best when it comes to page size. There are different factors to consider in order to come up with a suitable page size, such as page table, paging time, and its effect on the overall efficiency of the operating system.

**31) When designing the file structure for an operating system, what attributes are considered?**

Typically, the different attributes for a file structure are naming, identifier, supported file types, and location for the files, size, and level of protection.

**32) What is root partition?**

Root partition is where the operating system kernel is located. It also contains other potentially important system files that are mounted during boot time.

**33) What are device drivers?**

Device drivers provide a standard means of representing I/O devices that maybe manufactured by different companies. This prevents conflicts whenever such devices are incorporated in a systems unit.

**34) What are the primary functions of VFS?**

VFS, or Virtual File System, separate file system generic operations from their implementation by defining a clean VFS interface. It is based on a file-representation structure known as vnode, which contains a numerical designator needed to support network file systems.

**35) What are the different types of CPU registers in a typical operating**[**system design**](https://career.guru99.com/software-design-interview-questions/)**?**

– Accumulators  
– Index Registers  
– Stack Pointer  
– General Purpose Registers

**36) What is the purpose of an I/O status information?**

I/O status information provides information about which I/O devices are to be allocated for a particular process. It also shows which files are opened, and other I/O device state.

**37) What is multitasking?**

Multitasking is the process within an operating system that allows the user to run several applications at the same time. However, only one application is active at a time for user interaction, although some applications can run “behind the scene”.

**38) Explain pros and cons of a command line interface?**

A command line interface allows the user to type in commands that can immediately provide results. Many seasoned computer users are well accustomed to using the command line because they find it quicker and simpler.

However, the main problem with a command line interface is that users have to be familiar with the commands, including the switches and parameters that come with it. This is a downside for people who are not fond of memorizing commands.

**39) What is caching?**

Caching is the processing of utilizing a region of fast memory for a limited data and process. A cache memory is usually much efficient because of its high access speed.

**40) What is spooling?**

Spooling is normally associated with printing. When different applications want to send an output to the printer at the same time, spooling takes all of these print jobs into a disk file and queues them accordingly to the printer.

**41) What is an Assembler?**

An assembler acts as a translator for low-level language. Assembly codes written using mnemonic commands are translated by the Assembler into machine language.

**42) What are interrupts?**

Interrupts are part of a hardware mechanism that sends a notification to the CPU when it wants to gain access to a particular resource. An interrupt handler receives this interrupt signal and “tells” the processor to take action based on the interrupt request.

**43) What is GUI?**

GUI is short for Graphical User Interface. It provides users with an interface wherein actions can be performed by interacting with icons and graphical symbols. People find it easier to interact with the computer when in a GUI especially when using the mouse. Instead of having to remember and type commands, users click on buttons to perform a process.

**44) What is preemptive multitasking?**

Preemptive multitasking allows an operating system to switch between software programs. This, in turn, allows multiple programs to run without necessarily taking complete control over the processor and resulting in system crashes.

**45) Why partitioning and formatting is a prerequisite to installing an operating system?**

Partitioning and formatting create a preparatory environment on the drive so that the operating system can be copied and installed properly. This includes allocating space on the drive, designating a drive name, determining and creating the appropriate file system and structure.

**46) What is plumbing/piping?**

It is the process of using the output of one program as an input to another. For example, instead of sending the listing of a folder or drive to the main screen, it can be piped and sent to a file, or sent to the printer to produce a hard copy.

**47) What is NOS?**

NOS is short for Network Operating System. It is a specialized software that will allow a computer to communicate with other devices over the network, including file/folder sharing.

**48) Differentiate internal commands from external commands.**

Internal commands are built-in commands that are already part of the operating system. External commands are separate file programs that are stored in a separate folder or directory.

**49) Under DOS, what command will you type when you want to list down the files in a directory, and at the same time pause after every screen output?**  
**a) dir /w  
b) dir /p  
c) dir /s  
d) dir /w /p**

Answer: d) dir /w /p

**50) How would a file name EXAMPLEFILE.TXT appear when viewed under the DOS command console operating in Windows 98?**

The filename would appear as EXAMPL~1.TXT . The reason behind this is that filenames under this operating system are limited to 8 characters when working under DOS environment.

**51) What is a folder in Ubuntu?**

There is no concept of Folder in Ubuntu. Everything included in your hardware is a FILE.

**52) Explain why Ubuntu is safe and not affected by viruses?**

* It does not support malicious e-mails and contents, and before any e-mail is opened by users it will go through many security checks
* Ubuntu uses Linux, which is a super secure O.S system
* Unlike other O.S, countless Linux users can see the code at any time and can fix the problem if there is any
* Malware and viruses are coded to take advantage of the weakness in Windows

**53) Explain what is Unity in Ubuntu? How can you add new entries to the launcher?**

In Ubuntu, Unity is the default graphical shell.  On the left side of the Ubuntu, it introduces the launcher and Dash to start programs.

In order to add new entries to the launcher, you can create a file name like **.desktop** and then drag the file on the launcher.

**54) Explain the purpose of using a libaio package in Ubuntu?**

Libaio is Linux Kernel Asynchronous I/O (A/O).  A/O allows even a single application thread to overlap I/O operations with other processing, by providing an interface for submitting one or more I/O requests in one system call without waiting for completion.  And a separate interface to reap completed I/O operations associated with a given completion group.

**55) What is the use of behavior tab in Ubuntu?**

Through behaviors tab, you can make many changes on the appearance of the desktop

* Auto-hide the launcher: You can use this option to reveal the launcher when moving the pointer to the defined hot spot.
* Enable workspaces:  By checking this option, you can enable workspace
* Add show desktop icon to the launcher: This option is used to display the desktop icon on the launcher

**56) What is the meaning of “export” command in Ubuntu?**

Export is a command in Bash shell language. When you try to set a variable, it is visible or exported to any subprocess started from that instance of bash.  The variable will not exist in the sub-process without the export command.

**57) Explain how you can reset Unity Configuration?**

To reset the unity configuration the simplest way to do is to hit open a Terminal or hit Atl-F2  and run the command # unity –reset

**58) Explain how to access Terminal?**

To access terminal, you have to go under Application **Menu -> Accessories -> Terminal.**

JAVA:

**Q #1) What is JAVA?**

**Answer:** Java is a high-level programming language and is platform-independent.

Java is a collection of objects. It was developed by Sun Microsystems. There are a lot of applications, websites, and games that are developed using Java.

**Q #2) What are the features of JAVA?**

**Answer: Features of Java are as follows:**

* **OOP concepts**
  + Object-oriented
  + Inheritance
  + Encapsulation
  + Polymorphism
  + Abstraction
* **Platform independent:** A single program works on different platforms without any modification.
* **High Performance:** JIT (Just In Time compiler) enables high performance in Java. JIT converts the bytecode into machine language and then JVM starts the execution.
* **Multi-threaded:** A flow of execution is known as a Thread. JVM creates a thread which is called the main thread. The user can create multiple threads by extending the thread class or by implementing the Runnable interface.

**Q #3) How does Java enable high performance?**

**Answer:** Java uses Just In Time compiler to enable high performance. It is used to convert the instructions into bytecodes.

**Q #4) Name the Java IDE’s?**

**Answer:** Eclipse and NetBeans are the IDE’s of JAVA.

**Q #5) What do you mean by Constructor?**

**Answer: Constructor can be explained in detail with enlisted points:**

* When a new object is created in a program a constructor gets invoked corresponding to the class.
* The constructor is a method which has the same name as the class name.
* If a user doesn’t create a constructor implicitly a default constructor will be created.
* The constructor can be overloaded.
* If the user created a constructor with a parameter then he should create another constructor explicitly without a parameter.

**Q #6) What is meant by the Local variable and the Instance variable?**

**Answer:**

**Local variables** are defined in the method and scope of the variables that exist inside the method itself.

**Instance variable** is defined inside the class and outside the method and the scope of the variables exists throughout the class.

**Q #7) What is a Class?**

**Answer:** All Java codes are defined in a Class. It has variables and methods.

**Variables**are attributes which define the state of a class.

**Methods** are the place where the exact business logic has to be done. It contains a set of statements (or) instructions to satisfy the particular requirement.

**Example:**

|  |
| --- |
| public class Addition{ //Class name declaration  int a = 5; //Variable declaration  int b= 5;  public void add(){ //Method declaration  int c = a+b;  }  } |

**Q #8) What is an Object?**

**Answer:** An instance of a class is called an object. The object has state and behavior.

Whenever the JVM reads the “new()” keyword then it will create an instance of that class.

**Example:**

|  |
| --- |
| public class Addition{  public static void main(String[] args){  Addion add = new Addition();//Object creation  }  } |

The above code creates the object for the Addition class.

**Q #9)What are the OOPs concepts?**

**Answer: OOPs concepts include:**

* Inheritance
* Encapsulation
* Polymorphism
* Abstraction
* Interface

***Suggested Read =>>***[***Top OOPs Interview Questions***](https://www.softwaretestinghelp.com/oops-interview-questions-and-answers/)

**Q #10) What is Inheritance?**

**Answer:** Inheritance means one class can extend to another class. So that the codes can be reused from one class to another class. The existing class is known as the Super class whereas the derived class is known as a sub class.

**Example:**

|  |
| --- |
| Super class:  public class Manupulation(){  }  Sub class:  public class Addition extends Manipulation(){  } |

Inheritance is only applicable to the public and protected members only. Private members can’t be inherited.

**Q #11) What is Encapsulation?**

**Answer: Purpose of Encapsulation:**

* Protects the code from others.
* Code maintainability.

**Example:**

We are declaring ‘a’ as an integer variable and it should not be negative.

|  |
| --- |
| public class Addition(){  int a=5;  } |

If someone changes the exact variable as “***a = -5”***then it is bad.

**In order to overcome the problem we need to follow the steps below:**

* We can make the variable private or protected.
* Use public accessor methods such as set<property> and get<property>.

**So that the above code can be modified as:**

|  |
| --- |
| public class Addition(){  private int a = 5; //Here the variable is marked as private  } |

**The code below shows the getter and setter.**

Conditions can be provided while setting the variable.

|  |
| --- |
| get A(){  }  set A(int a){  if(a&gt;0){// Here condition is applied  .........  }  } |

For encapsulation, we need to make all the instance variables private and create setter and getter for those variables. Which in turn will force others to call the setters rather than access the data directly.

**Q #12) What is Polymorphism?**

**Answer:** Polymorphism means many forms.

A single object can refer to the super-class or sub-class depending on the reference type which is called polymorphism.

**Example:**

|  |
| --- |
| Public class Manipulation(){ //Super class  public void add(){  }  }  public class Addition extends Manipulation(){ // Sub class  public void add(){  }  public static void main(String args[]){  Manipulation addition = new Addition();//Manipulation is reference type and Addition is reference type  addition.add();  }  } |

Using the Manipulation reference type we can call the Addition class “add()” method. This ability is known as Polymorphism. Polymorphism is applicable for **overriding**and not for **overloading**.

**Q #13) What is meant by Method Overriding?**

**Answer: Method overriding happens if the sub-class method satisfies the below conditions with the Super-class method:**

* Method name should be the same
* The argument should be the same
* Return type should also be the same

The key benefit of overriding is that the Sub-class can provide some specific information about that sub-class type than the super-class.

**Example:**

|  |
| --- |
| public class Manipulation{ //Super class  public void add(){  ………………  }  }    Public class Addition extends Manipulation(){  Public void add(){  ………..  }  Public static void main(String args[]){  Manipulation addition = new Addition(); //Polimorphism is applied  addition.add(); // It calls the Sub class add() method  }  } |

**addition.add()**method calls the add() method in the Sub-class and not the parent class. So it overrides the Super-class method and is known as Method Overriding.

**Q #14) What is meant by Overloading?**

**Answer:** Method overloading happens for different classes or within the same class.

**For method overloading, sub-class method should satisfy the below conditions with the Super-class method (or) methods in the same class itself:**

* Same method name
* Different argument types
* There may be different return types

**Example:**

|  |
| --- |
| public class Manipulation{ //Super class  public void add(String name){ //String parameter  ………………  }  }    Public class Addition extends Manipulation(){  Public void add(){//No Parameter  ………..  }  Public void add(int a){ //integer parameter    }  Public static void main(String args[]){  Addition addition = new Addition();  addition.add();  }  } |

Here the add() method has different parameters in the Addition class is overloaded in the same class as with the super-class.

**Note:** Polymorphism is not applicable for method overloading.

**Q #15) What is meant by Interface?**

**Answer:** Multiple inheritances cannot be achieved in java. To overcome this problem the Interface concept is introduced.

An interface is a template which has only method declarations and not the method implementation.

**Example:**

|  |
| --- |
| Public abstract interface IManupulation{ //Interface declaration  Public abstract void add();//method declaration  public abstract void subtract();  } |

* All the methods in the interface are internally **public abstract void**.
* All the variables in the interface are internally **public static final** that is constants.
* Classes can implement the interface and not extends.
* The class which implements the interface should provide an implementation for all the methods declared in the interface.

|  |
| --- |
| public class Manupulation implements IManupulation{ //Manupulation class uses the interface  Public void add(){  ……………  }  Public void subtract(){  …………….  }  } |

**Q #16) What is meant by Abstract class?**

**Answer:** We can create the Abstract class by using the “Abstract” keyword before the class name. An abstract class can have both “Abstract” methods and “Non-abstract” methods that are a concrete class.

**Abstract method:**

The method which has only the declaration and not the implementation is called the abstract method and it has the keyword called “abstract”. Declarations ends with a semicolon.

**Example:**

|  |
| --- |
| public abstract class Manupulation{  public abstract void add();//Abstract method declaration  Public void subtract(){  }  } |

* An abstract class may have a non- abstract method also.
* The concrete Subclass which extends the Abstract class should provide the implementation for abstract methods.

**Q #17) Difference between Array and Array List.**

**Answer:** **The Difference between Array and Array List can be understood from the table below:**

| **Array** | **Array List** |
| --- | --- |
| Size should be given at the time of array declaration.  String[] name = new String[2] | Size may not be required. It changes the size dynamically.  ArrayList name = new ArrayList |
| To put an object into array we need to specify the index.  name[1] = “book” | No index required.  name.add(“book”) |
| Array is not type parameterized | ArrayList in java 5.0 are parameterized.  Eg: This angle bracket is a type parameter which means a list of String. |

**Q #18) Difference between String, String Builder, and String Buffer.**

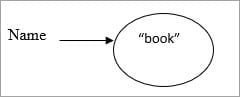
**Answer:**

**String:** String variables are stored in a “constant string pool”. Once the string reference changes the old value that exists in the “constant string pool”, it cannot be erased.

**Example:**

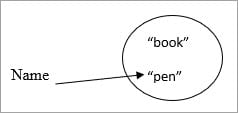
String name = “book”;

**Constant string pool**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Constant-string-pool.jpg).

If the name-value has changed from “book” to “pen”.

**Constant string pool**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Constant-string-pools.jpg)

Then the older value remains in the constant string pool.

**String Buffer:**

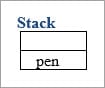
* Here string values are stored in a stack. If the values are changed then the new value replaces the older value.
* The string buffer is synchronized which is thread-safe.
* Performance is slower than the String Builder.

**Example:**

String Buffer name =”book”;

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Stack.jpg)

Once the name value has been changed to “pen” then the “book” is erased in the stack.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Stack1.jpg)

**String Builder:**

This is the same as String Buffer except for the String Builder which is not threaded safely that is not synchronized. So obviously the performance is fast.

**Q #19) Explain about Public and Private access specifiers.**

**Answer:** Methods and instance variables are known as members.

**Public:**

Public members are visible in the same package as well as the outside package that is for other packages.

Public members of Class A are visible to Class B (same package) as well as Class C (different packages).

**Private:**

Private members are visible in the same class only and not for the other classes in the same package as well as classes in the outside packages.

Private members in class A are visible only in that class. It is invisible for class  B as well as class C.

**Q #20) Difference between Default and Protected access specifiers.**

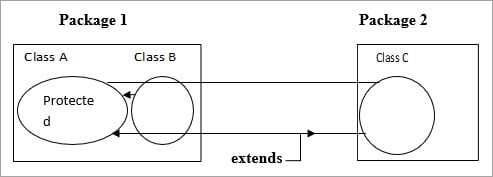
**Answer:**

**Default:**Methods and variables declared in a class without any access specifiers are called default.

Default members in Class A are visible to the other classes which are inside the package and invisible to the classes which are outside the package.

So Class A members are visible to Class B and invisible to Class C.

**Protected:**

**[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Protected.jpg)             .**

Protected is the same as Default but if a class extends then it is visible even if it is outside the package.

Class A members are visible to Class B because it is inside the package. For Class C it is invisible but if Class C extends Class A then the members are visible to Class C even if it is outside the package.

**Q #21) Difference between HashMap and HashTable.**

**Answer:** **The difference between HashMap and HashTable can be seen below:**

| **HashMap** | **HashTable** |
| --- | --- |
| Methods are not synchronized | Key methods are synchronized |
| Not thread safety | Thread safety |
| Iterator is used to iterate the values | Enumerator is used to iterate the values |
| Allows one null key and multiple null values | Doesn’t allow anything that is null |
| Performance is high than HashTable | Performance is slow |

**Q #22) Difference between HashSet and TreeSet.**

**Answer:** **The difference between HashSet and TreeSet can be seen below:**

| **HashSet** | **TreeSet** |
| --- | --- |
| Inserted elements are in random order | Maintains the elements in the sorted order |
| Can able to store null objects | Couldn’t store null objects |
| Performance is fast | Performance is slow |

**Q #23) Difference between Abstract class and Interface.**

**Answer: The differences between Abstract Class and Interface are as follows:**

**Abstract Class:**

* Abstract classes have a default constructor and it is called whenever the concrete subclass is instantiated.
* It contains Abstract methods as well as Non-Abstract methods.
* The class which extends the Abstract class shouldn’t require the implementation of all the methods, only Abstract methods need to be implemented in the concrete sub-class.
* Abstract class contains instance variables.

**Interface:**

* It doesn’t have any constructor and couldn’t be instantiated.
* The abstract method alone should be declared.
* Classes that implement the interface should provide the implementation for all the methods.
* The interface contains only constants.

**Q** **#24)  What is the meaning of Collections in Java?**

**Answer:** Collection is a framework that is designed to store the objects and manipulate the design to store the objects.

**Collections are used to perform the following operations:**

* Searching
* Sorting
* Manipulation
* Insertion
* Deletion

A group of objects is known as collections. All the classes and interfaces for collecting are available in Java util package.

**Q #25) What are all the Classes and Interfaces that are available in the collections?**

**Answer:** **Given below are the Classes and Interfaces that are available in Collections:**

**Interfaces:**

* Collection
* List
* Set
* Map
* Sorted Set
* Sorted Map
* Queue

**Classes:**

* Lists:
* Array List
* Vector
* Linked List

**Sets:**

* Hash set
* Linked Hash Set
* Tree Set

**Maps:**

* Hash Map
* Hash Table
* TreeMap
* Linked Hashed Map

**Queue:**

* Priority Queue

**Q #26) What is meant by Ordered and Sorted in collections?**

**Answer:**

**Ordered:**It means the values that are stored in a collection is based on the values that are added to the collection. So we can iterate the values from the collection in a specific order.

**Sorted:**Sorting mechanisms can be applied internally or externally so that the group of objects sorted in a particular collection is based on the properties of the objects.

**Q #27) Explain the different lists available in the collection.**

**Answer:**Values added to the list are based on the index position and it is ordered by index position. Duplicates are allowed.

**The types of Lists are:**

**a) Array List:**

* Fast iteration and fast Random Access.
* It is an ordered collection (by index) and not sorted.
* It implements the Random Access Interface.

**Example:**

|  |
| --- |
| public class Fruits{  public static void main (String [ ] args){  ArrayList &lt;String&gt;names=new ArrayList &lt;String&gt;();  names.add (“apple”);  names.add (“cherry”);  names.add (“kiwi”);  names.add (“banana”);  names.add (“cherry”);  System.out.println (names);  }  } |

**Output:**

[Apple, cherry, kiwi, banana, cherry]

From the output, Array List maintains the insertion order and it accepts the duplicates. But it’s not sorted.

**b) Vector:**

It is the same as Array List.

* Vector methods are synchronized.
* Thread safety.
* It also implements Random Access.
* Thread safety usually causes a performance hit.

**Example:**

|  |
| --- |
| public class Fruit {  public static void main (String [ ] args){  Vector &lt;String&gt; names = new Vector &lt;String&gt; ( );   names.add (“cherry”);  names.add (“apple”);  names.add (“banana”);  names.add (“kiwi”);  names.add (“apple”);  System.out.println (“names”);  }  } |

**Output:**

[cherry,apple,banana,kiwi,apple]

Vector also maintains the insertion order and accepts the duplicates.

**c) Linked List:**

* Elements are doubly linked to one another.
* Performance is slower than the Array list.
* Good choice for insertion and deletion.
* In Java 5.0 it supports common queue methods peek( ), Pool ( ), Offer ( ) etc.

**Example:**

|  |
| --- |
| public class Fruit {  public static void main (String [ ] args){  Linkedlist &lt;String&gt; names = new linkedlist &lt;String&gt; ( ) ;  names.add(“banana”);  names.add(“cherry”);  names.add(“apple”);  names.add(“kiwi”);  names.add(“banana”);  System.out.println (names);  }  } |

**Output:**

[ banana,cherry,apple,kiwi,banana]

Maintains the insertion order and accepts the duplicates.

**Q #28) Explain about Set and their types in a collection.**

**Answer:** Set cares about uniqueness. It doesn’t allow duplications. Here “equals ( )” method is used to determine whether two objects are identical or not.

**a) Hash Set:**

* Unordered and unsorted.
* Uses the hash code of the object to insert the values.
* Use this when the requirement is “no duplicates and don’t care about the order”.

**Example:**

|  |
| --- |
| public class Fruit {  public static void main (String[ ] args){  HashSet&lt;String&gt; names = new HashSet &lt;=String&gt;( ) ;  names.add(“banana”);  names.add(“cherry”);  names.add(“apple”);  names.add(“kiwi”);  names.add(“banana”);  System.out.println (names);  }  } |

**Output:**

[banana, cherry, kiwi, apple]

It doesn’t follow any insertion order. Duplicates are not allowed.

**b) Linked Hash set:**

* An ordered version of the hash set is known as Linked Hash Set.
* Maintains a doubly-Linked list of all the elements.
* Use this when an iteration order is required.

**Example:**

|  |
| --- |
| public class Fruit {  public static void main (String[ ] args){  LinkedHashSet&lt;String&gt;; names = new LinkedHashSet &lt;String&gt;( ) ;   names.add(“banana”);   names.add(“cherry”);   names.add(“apple”);   names.add(“kiwi”);   names.add(“banana”);   System.out.println (names);   }  } |

**Output:**

[banana, cherry, apple, kiwi]

It maintains the insertion order in which they have been added to the Set. Duplicates are not allowed.

**c) Tree Set:**

* It is one of the two sorted collections.
* Uses the “Read-Black” tree structure and guarantees that the elements will be in ascending order.
* We can construct a tree set with the constructor by using a comparable (or) comparator.

**Example:**

|  |
| --- |
| public class Fruits{  public static void main (String[ ]args) {  Treeset&lt;String&gt; names= new TreeSet&lt;String&gt;( ) ;  names.add(“cherry”);  names.add(“banana”);  names.add(“apple”);  names.add(“kiwi”);  names.add(“cherry”);  System.out.println(names);  }  } |

**Output:**

[apple, banana, cherry, kiwi]

TreeSet sorts the elements in ascending order. And duplicates are not allowed.

**Q #29) Explain about Map and its types.**

**Answer: Map** cares about the unique identifier. We can map a unique key to a specific value. It is a key/value pair. We can search a value, based on the key. Like the set, the map also uses the “equals ( )” method to determine whether two keys are the same or different.

**Map is of following types:**

**a) Hash Map:**

* Unordered and unsorted map.
* Hashmap is a good choice when we don’t care about the order.
* It allows one null key and multiple null values.

**Example:**

|  |
| --- |
| Public class Fruit{  Public static void main(String[ ] args){  HashMap&lt;Sting,String&gt; names =new HashMap&lt;String,String&gt;( );  names.put(“key1”,“cherry”);  names.put (“key2”,“banana”);  names.put (“key3”,“apple”);  names.put (“key4”,“kiwi”);  names.put (“key1”,“cherry”);  System.out.println(names);  }   } |

**Output:**

{key2 =banana, key1=cherry, key4 =kiwi, key3= apple}

Duplicate keys are not allowed in Map.

It doesn’t maintain any insertion order and is unsorted.

**b) Hash Table:**

* Like the vector key, methods of the class are synchronized.
* Thread safety and therefore slows the performance.
* It doesn’t allow anything that is null.

**Example:**

|  |
| --- |
| public class Fruit{  public static void main(String[ ]args){  Hashtable&lt;Sting,String&gt; names =new Hashtable&lt;String,String&gt;( );  names.put(“key1”,“cherry”);  names.put(“key2”,“apple”);  names.put(“key3”,“banana”);  names.put(“key4”,“kiwi”);  names.put(“key2”,“orange”);  System.out.println(names);  }   } |

**Output:**

{key2=apple, key1=cherry,key4=kiwi, key3=banana}

Duplicate keys are not allowed.

**c) Linked Hash Map:**

* Maintains insertion order.
* Slower than Hash map.
* I can expect a faster iteration.

**Example:**

|  |
| --- |
| public class Fruit{  public static void main(String[ ] args){  LinkedHashMap&lt;Sting,String&gt; names =new LinkedHashMap&lt;String,String&gt;( );   names.put(“key1”,“cherry”);   names.put(“key2”,“apple”);   names.put(“key3”,“banana”);   names.put(“key4”,“kiwi”);   names.put(“key2”,“orange”);   System.out.println(names);   }   } |

**Output:**

{key2=apple, key1=cherry,key4=kiwi, key3=banana}

Duplicate keys are not allowed.

**d) TreeMap:**

* Sorted Map.
* Like Tree set, we can construct a sort order with the constructor.

**Example:**

|  |
| --- |
| public class Fruit{  public static void main(String[ ]args){  TreeMap&lt;Sting,String&gt; names =new TreeMap&lt;String,String&gt;( );  names.put(“key1”,“cherry”);  names.put(“key2”,“banana”);  names.put(“key3”,“apple”);  names.put(“key4”,“kiwi”);  names.put(“key2”,“orange”);  System.out.println(names);  }  } |

**Output:**

{key1=cherry, key2=banana, key3 =apple, key4=kiwi}

It is sorted in ascending order based on the key. Duplicate keys are not allowed.

**Q #30) Explain the Priority Queue.**

**Answer: Queue Interface**

**Priority Queue:**Linked list class has been enhanced to implement the queue interface. Queues can be handled with a linked list. The purpose of a queue is “Priority-in, Priority-out”.

Hence elements are ordered either naturally or according to the comparator. The elements ordering represents their relative priority.

**Q #31) What is meant by Exception?**

**Answer:** An Exception is a problem that can occur during the normal flow of execution. A method can throw an exception when something wails at runtime. If that exception couldn’t be handled, then the execution gets terminated before it completes the task.

If we handled the exception, then the normal flow gets continued. Exceptions are a subclass of java.lang.Exception.

**Example for handling Exception:**

|  |
| --- |
| try{  //Risky codes are surrounded by this block  }catch(Exception e){  //Exceptions are caught in catch block  } |

**Q #32) What are the types of Exceptions?**

**Answer:** There are two types of Exceptions. They are explained below in detail.

**a) Checked Exception:**

These exceptions are checked by the compiler at the time of compilation. Classes that extend Throwable class except Runtime exception and Error are called checked Exception.

Checked Exceptions must either declare the exception using throws keyword (or) surrounded by appropriate try/catch.

**For Example,** ClassNotFound Exception

**b) Unchecked Exception:**

These exceptions are not checked during the compile time by the compiler.  The compiler doesn’t force to handle these exceptions. **It includes:**

* Arithmetic Exception
* ArrayIndexOutOfBounds Exception

**Q #33) What are the different ways to handle exceptions?**

**Answer:** **Two different ways to handle exceptions are explained below:**

**a) Using try/catch:**

The risky code is surrounded by try block. If an exception occurs, then it is caught by the catch block which is followed by the try block.

**Example:**

|  |
| --- |
| class Manipulation{  public static void main(String[] args){  add();  }  Public void add(){  try{  addition();  }catch(Exception e){  e.printStacktrace();  }  }  } |

**b) By declaring throws keyword:**

At the end of the method, we can declare the exception using throws keyword.

**Example:**

|  |
| --- |
| class Manipulation{  public static void main(String[] args){  add();  }  public void add() throws Exception{  addition();  }  } |

**Q #34) What are the advantages of Exception handling?**

**Answer: The advantages are as follows:**

* The normal flow of the execution won’t be terminated if an exception gets handled
* We can identify the problem by using catch declaration

**Q #35) What are the Exception handling keywords in Java?**

**Answer: Enlisted below are the two Exception Handling Keywords:**

**a) try:**

When a risky code is surrounded by a try block. An exception occurring in the try block is caught by a catch block. Try can be followed either by catch (or) finally (or) both. But any one of the blocks is mandatory.

**b) catch:**

This is followed by a try block. Exceptions are caught here.

**c) finally:**

This is followed either by try block (or) catch block. This block gets executed regardless of an exception. So generally clean up codes are provided here.

**Q #36) Explain about Exception Propagation.**

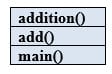
**Answer:** Exception is first thrown from the method which is at the top of the stack. If it doesn’t catch, then it pops up the method and moves to the previous method and so on until they are got.

This is called Exception propagation.

**Example:**

|  |
| --- |
| public class Manipulation{  public static void main(String[] args){  add();  }  public void add(){  addition();  } |

**From the above example, the stack looks like as shown below:**

**[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Stack-Example.jpg)**

If an exception occurs in the **addition()** method is not caught, then it moves to the method **add()**. Then it is moved to the **main()** method and then it will stop the flow of execution. It is called Exception Propagation.

**Q #37) What is the final keyword in Java?**

**Answer:**

**Final variable:**Once a variable is declared as final, then the value of the variable could not be changed. It is like a constant.

**Example:**

final int = 12;

**Final method:**A final keyword in a method, couldn’t be overridden. If a method is marked as a final, then it can’t be overridden by the subclass.

**Final class:**If a class is declared as final, then the class couldn’t be subclassed. No class can extend the final class.

**Q #38) What is a Thread?**

**Answer:**In Java, the flow of execution is called Thread. Every java program has at least one thread called the main thread, the main thread is created by JVM. The user can define their own threads by extending the Thread class (or) by implementing the Runnable interface. Threads are executed concurrently.

**Example:**

|  |
| --- |
| public static void main(String[] args){//main thread starts here  } |

**Q #39) How do you make a thread in Java?**

**Answer:**There are two ways available to make a thread.

**a) Extend Thread class:**Extending a Thread class and override the run method. The thread is available in java.lang.thread.

**Example:**

|  |
| --- |
| Public class Addition extends Thread {  public void run () {  }  } |

The disadvantage of using a thread class is that we cannot extend any other classes because we have already extended the thread class. We can overload the run () method in our class.

**b) Implement Runnable interface:**Another way is by implementing the runnable interface. For that, we should provide the implementation for the run () method which is defined in the interface.

**Example:**

|  |
| --- |
| Public class Addition implements Runnable {  public void run () {  }  } |

**Q #40) Explain about join () method.**

**Answer:** Join () method is used to join one thread with the end of the currently running thread.

**Example:**

|  |
| --- |
| public static void main (String[] args){  Thread t = new Thread ();  t.start ();  t.join ();  } |

Based on the above code, the main thread has started the execution. When it reaches the code ***t.start()*** then ‘thread t’ starts the own stack for the execution. JVM switches between the main thread and ‘thread t’.

Once it reaches the code ***t.join()*** then ‘thread t’ alone is executed and completes its task, then only the main thread starts the execution.

It is a non-static method. The Join () method has an overloaded version. So we can mention the time duration in join () method also “.s”.

**Q #41) What does the yield method of the Thread class do?**

**Answer:** A yield () method moves the currently running thread to a runnable state and allows the other threads for execution. So that equal priority threads have a chance to run. It is a static method. It doesn’t release any lock.

Yield () method moves the thread back to the Runnable state only, and not the thread to sleep (), wait () (or) block.

**Example:**

|  |
| --- |
| public static void main (String[] args){  Thread t = new Thread ();  t.start ();  }  public void run(){  Thread.yield();  }  } |

**Q #42) Explain about wait () method.**

**Answer: wait ()** method is used to make the thread to wait in the waiting pool. When the wait () method is executed during a thread execution then immediately the thread gives up the lock on the object and goes to the waiting pool. Wait () method tells the thread to wait for a given amount of time.

Then the thread will wake up after notify () (or) notify all () method is called.

Wait() and the other above-mentioned methods do not give the lock on the object immediately until the currently executing thread completes the synchronized code. It is mostly used in synchronization.

**Example:**

|  |
| --- |
| public static void main (String[] args){  Thread t = new Thread ();  t.start ();  Synchronized (t) {  Wait();  }  } |

**Q #43) Difference between notify() method and notifyAll() method in Java.**

**Answer: The differences between notify() method and notifyAll() method are enlisted below:**

| **notify()** | **notifyAll()** |
| --- | --- |
| This method is used to send a signal to wake up a single thread in the waiting pool. | This method sends the signal to wake up all the threads in a waiting spool. |

**Q #44) How to stop a thread in java? Explain about sleep () method in a thread?**

**Answer:** **We can stop a thread by using the following thread methods:**

* Sleeping
* Waiting
* Blocked

**Sleep:**Sleep () method is used to sleep the currently executing thread for the given amount of time. Once the thread is wake up it can move to the runnable state. So sleep () method is used to delay the execution for some period.

It is a static method.

**Example:**

**Thread. Sleep (2000)**

So it delays the thread to sleep 2 milliseconds. Sleep () method throws an uninterrupted exception, hence we need to surround the block with try/catch.

|  |
| --- |
| public class ExampleThread implements Runnable{  public static void main (String[] args){  Thread t = new Thread ();  t.start ();  }  public void run(){  try{  Thread.sleep(2000);  }catch(InterruptedException e){  }  } |

**Q #45) When to use the Runnable interface Vs Thread class in Java?**

**Answer:** If we need our class to extend some other classes other than the thread then we can go with the runnable interface because in java we can extend only one class.

If we are not going to extend any class then we can extend the thread class.

**Q #46) Difference between start() and run() method of thread class.**

**Answer:** Start() method creates a new thread and the code inside the run () method is executed in the new thread. If we directly called the run() method then a new thread is not created and the currently executing thread will continue to execute the run() method.

**Q #47) What is Multi-threading?**

**Answer:** Multiple threads are executed simultaneously. Each thread starts its own stack based on the flow (or) priority of the threads.

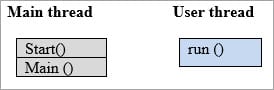
**Example Program:**

|  |
| --- |
| public class MultipleThreads implements Runnable  {  public static void main (String[] args){//Main thread starts here  Runnable r = new runnable ();  Thread t=new thread ();  t.start ();//User thread starts here  Addition add=new addition ();  }  public void run(){  go();  }//User thread ends here  } |

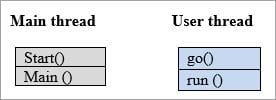
On the 1st line execution, JVM calls the main method and the main thread stack looks as shown below.

Once the execution reaches, **t.start ()**line then a new thread is created and the new stack for the thread is also created. Now JVM switches to the new thread and the main thread are back to the runnable state.

The two stacks look as shown below.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Thread2.jpg)

Now, the user thread executed the code inside the run() method.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2017/10/Thread3.jpg)

Once the run() method has completed, then JVM switches back to the main thread and the user thread has completed the task and the stack was disappeared.

JVM switches between each thread until both the threads are completed. This is called Multi-threading.

**Q #48) Explain the thread life cycle in Java.**

**Answer:** **Thread has the following states:**

* New
* Runnable
* Running
* Non-runnable (Blocked)
* Terminated

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/01/Thread-Life-Cycle-in-Java.jpg)

* **New:**In New state, a Thread instance has been created but start () method is not yet invoked. Now the thread is not considered alive.
* **Runnable**: The Thread is in the runnable state after the invocation of the start () method, but before the run () method is invoked. But a thread can also return to the runnable state from waiting/sleeping. In this state, the thread is considered alive.
* **Running**: The thread is in a running state after it calls the run () method. Now the thread begins the execution.
* **Non-Runnable**(Blocked): The thread is alive but it is not eligible to run. It is not in the runnable state but also, it will return to the runnable state after some time. **Example:** wait, sleep, block.
* **Terminated**: Once the run method is completed then it is terminated. Now the thread is not alive.

**Q #49) What is Synchronization?**

**Answer:** Synchronization makes only one thread to access a block of code at a time. If multiple threads accesses the block of code, then there is a chance for inaccurate results at the end. To avoid this issue, we can provide synchronization for the sensitive block of codes.

The synchronized keyword means that a thread needs a key in order to access the synchronized code.

Locks are per objects. Every Java object has a lock. A lock has only one key. A thread can access a synchronized method only if the thread can get the key to the objects to lock.

For this, we use the “Synchronized” keyword.

**Example:**

|  |
| --- |
| public class ExampleThread implements Runnable{   public static void main (String[] args){   Thread t = new Thread ();   t.start ();   }   public void run(){   synchronized(object){   {   }  } |

**Q #50) What is the disadvantage of Synchronization?**

**Ans:** Synchronization is not recommended to implement all the methods. Because if one thread accesses the synchronized code then the next thread should have to wait. So it makes a slow performance on the other end.

**Q #51) What is meant by Serialization?**

**Answer:** Converting a file into a byte stream is known as Serialization. The objects in the file are converted to bytes for security purposes. For this, we need to implement a java.io.Serializable interface. It has no method to define.

Variables that are marked as transient will not be a part of the serialization. So we can skip the serialization for the variables in the file by using a transient keyword.

**Q #52) What is the purpose of a transient variable?**

**Answer:** Transient variables are not part of the serialization process. During deserialization, the values of the transient variables are set to the default value. It is not used with static variables.

**Example:**

transient int numbers;

**Q #53) Which methods are used during the Serialization and Deserialization process?**

**Answer:** ObjectOutputStream and ObjectInputStream classes are higher level java.io. package. We will use them with lower level classes FileOutputStream and FileInputStream.

ObjectOutputStream.writeObject**—->**Serialize the object and write the serialized object to a file.

ObjectInputStream.readObject **—>** Reads the file and deserializes the object.

To be serialized, an object must implement the serializable interface. If superclass implements Serializable, then the subclass will automatically be serializable.

**Q #54) What is the purpose of a Volatile Variable?**

**Answer:** Volatile variable values are always read from the main memory and not from thread’s cache memory. This is used mainly during synchronization. It is applicable only for variables.

**Example:**

volatile int number;

**Q #55) Difference between Serialization and Deserialization in Java.**

**Answer:** **These are the differences between serialization and deserialization in java:**

| **Serialization** | **Deserialization** |
| --- | --- |
| Serialization is the process which is used to convert the objects into byte stream | Deserialization is the opposite process of serialization where we can get the objects back from the byte stream. |
| An object is serialized by writing it an ObjectOutputStream. | An object is deserialized by reading it from an ObjectInputStream. |

**Q #56) What is SerialVersionUID?**

**Answer:** Whenever an object is Serialized, the object is stamped with a version ID number for the object class. This ID is called the  SerialVersionUID. This is used during deserialization to verify that the sender and receiver that are compatible with the Serialization.