

Home    Verbal Ability    Aptitude    Reasoning    Programs    Java    JSP    Android

# Verbal Ability



## Verbal Ability

Verbal ability is an important section of most of the competitive exams. It contains different types of questions which are intended to judge abilities in word power, sentence correction, spotting-errors-2 and verbal reasoning.

So, to score good marks in verbal ability, the candidates need to have a good vocabulary and a strong command of English like in-depth knowledge of grammar, adjectives, tenses, articles and more.

Let us see, understand and practice the most important verbal ability topics or chapters with most frequently asked verbal ability solved questions.

### Table of Contents

- Spotting Errors
- Antonyms
- Synonyms
- Spellings
- Selecting Words
- Sentence Formation
- Ordering of Words
- Sentence Correction
- Ordering of Sentences
- Paragraph Formation
- Closet Test
- Comprehension
- One Word Substitutes
- Idioms and Phrases
- Change of Voice
- Change of Speech
- Verbal Analogies
- Articles
- Prepositions
- Adjectives

## 1) Spotting Errors

- 1)
- A. It is difficult
  - B. for anyone
  - C. to ran on a slippery surface
  - D. no error

**Show Answer**

**Workspace**

[Spotting Errors](#)

[Spotting Errors 2](#)

[Spotting Errors 3](#)

[Spotting Errors 4](#)

[Spotting Errors 5](#)

[Spotting Errors 6](#)

[Spotting Errors 7](#)

[Spotting Errors 8](#)

[Spotting Errors 9](#)

[Spotting Errors 10](#)

[Spotting Errors 11](#)

[Spotting Errors 12](#)

[Spotting Errors 13](#)

[Spotting Errors 14](#)

[Spotting Errors 15](#)

[Spotting Errors 16](#)

[Spotting Errors 17](#)

[Spotting Errors 18](#)[Spotting Errors 19](#)[Spotting Errors 20](#)

## 2) Antonyms

1) Choose one of the following options that means the opposite of the given word; Copious:

- A. Reverse
- B. Scarce
- C. Abundant
- D. Short

[Show Answer](#)[Workspace](#)[Antonyms](#)[Antonyms 1](#)[Antonyms 2](#)[Antonyms 3](#)[Antonyms 4](#)[Antonyms 5](#)[Antonyms 6](#)[Antonyms 7](#)[Antonyms 8](#)[Antonyms 9](#)[Antonyms 10](#)[Antonyms 11](#)[Antonyms 12](#)[Antonyms 13](#)

[Antonyms 14](#)[Antonyms 15](#)[Antonyms 16](#)[Antonyms 17](#)[Antonyms 18](#)[Antonyms 19](#)[Antonyms 20](#)

### 3) Synonyms

1) Brief

- A. Partial
- B. Short
- C. Limited
- D. Little

[Show Answer](#)[Workspace](#)[Synonyms](#)[Synonyms 2](#)[Synonyms 3](#)[Synonyms 4](#)[Synonyms 5](#)[Synonyms 6](#)

### 4) Spellings

1)

- A. Betterment
- B. Efficiant

C. Employd

D. Trainad

**Show Answer**

**Workspace**

Spellings

Spellings 2

Spellings 3

Spellings 4

Spellings 5

Spellings 6

## 5) Selecting Words

1) She lived \_\_\_\_ the expectations.

A. as per

B. up to

C. on

D. at

**Show Answer**

**Workspace**

Selecting Words

Selecting Words 2

Selecting Words 3

Selecting Words 4

Selecting Words 5

Selecting Words 6

## 6) Spotting Errors

- 1) 1.not 2.Maria 3.run 4.did 5.away
- A. 24153  
B. 24135  
C. 21354  
D. 41235

**Show Answer**

**Workspace**

Sentence Formation

Sentence Formation 2

Sentence Formation 3

Sentence Formation 4

Sentence Formation 5

Sentence Formation 6

## 7) Ordering of Words

- 1) A month after colleges opened.....

A. 35% seats reserved under OBC

B. in private colleges

C. for the next academic year

D. remain vacant

A. CBDA

B. CADB

C. ADBC

D. BADC

**Show Answer**

**Workspace**

Ordering of Words

Ordering of Words 2

### Ordering of Words 3

### Ordering of Words 4

### Ordering of Words 5

### Ordering of Words 6

## 8) Sentence Correction

1) He was very tired as he is working since 6 O' clock in the morning.

- A. he was working
- B. he had been working
- C. he has been working
- D. he will be working

**Show Answer**

**Workspace**

### Sentence Correction

### Sentence Correction 2

### Sentence Correction 3

### Sentence Correction 4

### Sentence Correction 5

### Sentence Correction 6

## 9) Ordering of Sentences

1)

Rishikesh is a beautiful hill station located beside the Himalayas

P: There are many ancient temples

Q: in the North of India

R: along the banks of the sacred river the Ganges

S: that flows through the city.

Furthermore, the city is also known for many famous yoga ashrams.

- A. SRPQ
- B. QPRS
- C. PQSR
- D. RSPQ

[Show Answer](#)[Workspace](#)[Ordering of Sentences](#)[Ordering of Sentences 2](#)[Ordering of Sentences 3](#)

## 10) Ordering of Sentences

1)

1. Maria reached office at 10 O' clock after sending the money.
2. Maria's parents needed some money.
3. After that, she spent almost half an hour at the Post Office.
4. So, she went to bank to withdraw some money.
5. However, she had no money with her.

Which of the following should come third in the paragraph?

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

[Show Answer](#)[Workspace](#)[Paragraph Formation](#)[Paragraph Formation 2](#)[Paragraph Formation 3](#)

## 11) Closet Test

1)

Today, doing business is not easy for businessmen, as they are not used to competition. In the past they were selling whatever ....(1).....produced and at their desired price. But, 2... the competition has increased over the years, customers started to ....3.....and choose. Imports have become....4....available and that too at cheaper .....5.....

### 1. Solve as per the direction given above

- A. he
- B. them
- C. it
- D. they

[Show Answer](#)

[Workspace](#)

### 2. Solve as per the direction given above

- A. with
- B. as
- C. when
- D. only

[Show Answer](#)

[Workspace](#)

### 3. Solve as per the direction given above

- A. pick
- B. select
- C. want
- D. take

[Show Answer](#)

[Workspace](#)

### 4. Solve as per the direction given above

- A. costly

- B. not
- C. easily
- D. hardly

[Show Answer](#)[Workspace](#)

### 5. Solve as per the direction given above

- A. inputs
- B. costs
- C. profits
- D. prices

[Show Answer](#)[Workspace](#)

Closet Test

Closet Test 2

Closet Test 3

## 12) Comprehension

1)

Harry who is a professional had a fearful dream. He found himself in a land where he saw some slug-like animals with tentacles living on human bodies. The people tolerated these creatures because after many years they would grow into bulls which then be used for transportation. Harry noticed that he himself was covered with these creatures and he woke up screaming.

### Question 1: In the dream, Harry found the creatures

- A. in his office
- B. in a different land
- C. in his kitchen
- D. in a different planet

[Show Answer](#)[Workspace](#)

**Question 2: what did the creatures look like?**

- A. slug-like animals with horns
- B. insects with wings
- C. insects with tentacles
- D. slug-like animals with tentacles

[Show Answer](#)[Workspace](#)**Question 3: Harry's dream was fearful because**

- A. It brought him face to face with elephants
- B. He found himself on a land full of snakes
- C. He forgets the way home
- D. He saw creatures feeding on human bodies

[Show Answer](#)[Workspace](#)**Question 4: The creatures will grow into bulls which then will be used for**

- A. digging
- B. hunting
- C. transportation
- D. flying

[Show Answer](#)[Workspace](#)**Question 5: Harry woke up**

- A. dancing
- B. screaming
- C. thinking
- D. singing

[Show Answer](#)[Workspace](#)**Comprehension**

## Comprehension 2

## Comprehension 3

### 13) One Word Substitutes

1) A person who travels on foot.

- A. pedestrian
- B. traveler
- C. walker
- D. health conscious

**Show Answer**

**Workspace**

One Word Substitutes

One Word Substitutes 2

One Word Substitutes 3

One Word Substitutes 4

One Word Substitutes 5

One Word Substitutes 6

One Word Substitutes 7

One Word Substitutes 8

One Word Substitutes 9

One Word Substitutes 10

One Word Substitutes 11

### 14) Idioms and Phrases

1) To cry wolf

- A. to speak loudly
- B. to eat like a wolf

- C. to get afraid
- D. to give a false alarm

[Show Answer](#)[Workspace](#)[Idioms and Phrases](#)[Idioms and Phrases 2](#)[Idioms and Phrases 3](#)[Idioms and Phrases 4](#)[Idioms and Phrases 5](#)[Idioms and Phrases 6](#)

## 15) Change of Voice

- 1) Her selection in the crew surprised Peter.
- A. Her selection in the crew was a big surprise.
  - B. Peter was surprised at her selection in the crew.
  - C. Her selection was a surprise.
  - D. Peter was surprised at her selection of the crew.

[Show Answer](#)[Workspace](#)[Change of Voice 2](#)[Change of Voice 3](#)[Change of Voice 4](#)[Change of Voice 5](#)[Change of Voice 6](#)

## 16) Change of Speech

- 1) They said, "We have lived in this city for many years."

- A. They said they have lived in this city for many years.
- B. They said that they had lived in this city for many years.
- C. They said they lived in this city for many years.
- D. They said they have been living in this city for many years.

[Show Answer](#)[Workspace](#)[Change of Speech](#)[Change of Speech 2](#)[Change of Speech 3](#)[Change of Speech 4](#)[Change of Speech 5](#)[Change of Speech 6](#)

## 17) Verbal Analogies

1) Liquid: Liter

- A. Hot: Cold
- B. Weight: Kilogram
- C. Movie: Entertainment
- D. Winter: Cold

[Show Answer](#)[Workspace](#)[Verbal Analogies](#)[Verbal Analogies 2](#)[Verbal Analogies 3](#)[Verbal Analogies 4](#)[Verbal Analogies 5](#)[Verbal Analogies 6](#)

## 18) Articles

1) ..... Indus River is the longest river in India.

- A. The
- B. An
- C. A
- D. None of the above

[Show Answer](#)[Workspace](#)[Articles](#)[Articles 2](#)[Articles 3](#)[Articles 4](#)[Articles 5](#)[Articles 6](#)[Articles 7](#)

## 19) Prepositions

1) They walked ..... the edge of the land.

- A. up to
- B. as far as
- C. Either A or B
- D. until

[Show Answer](#)[Workspace](#)[Prepositions](#)[Prepositions 2](#)[Prepositions 3](#)

[Prepositions 4](#)[Prepositions 5](#)[Prepositions 6](#)[Prepositions 7](#)[Prepositions 8](#)[Prepositions 9](#)[Prepositions 10](#)

## 20) Adjectives

1) On looking at the cloudless sky, Peter said, "It won't rain today."

- A. at
- B. cloudless
- C. sky
- D. today

[Show Answer](#)[Workspace](#)[Adjectives](#)[Adjectives 2](#)[Adjectives 3](#)[Adjectives 4](#)[Adjectives 5](#)[Adjectives 6](#)[Next →](#)

 [Youtube](#) For Videos Join Our Youtube Channel: [Join Now](#)

## Feedback

- Send your Feedback to [feedback@javatpoint.com](mailto:feedback@javatpoint.com)

## Help Others, Please Share



## Learn Latest Tutorials

<b>Splunk tutorial</b> Splunk	<b>SPSS tutorial</b> SPSS	<b>Swagger tutorial</b> Swagger	<b>T-SQL tutorial</b> Transact-SQL
<b>Tumblr tutorial</b> Tumblr	<b>React tutorial</b> ReactJS	<b>Regex tutorial</b> Regex	<b>Reinforcement learning tutorial</b> Reinforcement Learning
<b>R Programming tutorial</b> R Programming	<b>RxJS tutorial</b> RxJS	<b>React Native tutorial</b> React Native	<b>Python Design Patterns</b> Python Design Patterns
<b>Python Pillow tutorial</b> Python Pillow	<b>Python Turtle tutorial</b> Python Turtle	<b>Keras tutorial</b> Keras	

## Preparation

<b>Aptitude</b>	<b>Logical Reasoning</b>	<b>Verbal Ability</b>	<b>Interview Questions</b>
-----------------	--------------------------	-----------------------	----------------------------

## Reasoning

## Interview Questions



Company  
Interview  
Questions

Company Questions

## Trending Technologies



Artificial  
Intelligence  
  
Artificial  
Intelligence



AWS Tutorial  
AWS



Selenium  
tutorial  
  
Selenium



Cloud  
Computing  
  
Cloud Computing



Hadoop tutorial  
Hadoop



ReactJS  
Tutorial  
  
ReactJS



Data Science  
Tutorial  
  
Data Science



Angular 7  
Tutorial  
  
Angular 7



Blockchain  
Tutorial  
  
Blockchain



Git Tutorial  
Git



Machine  
Learning Tutorial  
  
Machine Learning



DevOps  
Tutorial  
  
DevOps

## B.Tech / MCA



DBMS tutorial  
DBMS



Data Structures  
tutorial  
  
Data Structures



DAA tutorial  
DAA



Operating  
System  
  
Operating System



Computer  
Network tutorial  
  
Computer Network



Compiler  
Design tutorial  
  
Compiler Design



Computer  
Organization and  
Architecture



Discrete  
Mathematics  
Tutorial

 Ethical Hacking Ethical Hacking	 Computer Graphics Tutorial Computer Graphics	 Software Engineering Software Engineering	 Discrete Mathematics Web Technology
 Cyber Security tutorial Cyber Security	 Automata Tutorial Automata	 C Language tutorial C Programming	 C++ tutorial C++
 Java tutorial Java	 .Net Framework tutorial .Net	 Python tutorial Python	 List of Programs Programs
 Control Systems tutorial Control System	 Data Mining Tutorial Data Mining	 Data Warehouse Tutorial Data Warehouse	



# Commonly Asked Data Structure Interview Questions

Last Updated : 16 Feb, 2024

In the world of tech interviews, knowing about data structures is super important for candidates aiming for jobs in computer science. Being good at data structures shows you can solve problems well and make programs run faster. This article is packed with top interview questions and answers about data structures. It's here to help you get ready for tough technical interviews.

## Table of Content

- [Commonly Asked Data Structure Interview Questions on Array](#)
- [Commonly Asked Data Structure Interview Questions on Linked List](#)
- [Commonly Asked Data Structure Interview Questions on Stack](#)
- [Commonly Asked Data Structure Interview Questions on Queue](#)
- [Commonly Asked Data Structure Interview Questions on Heap Data Structure](#)
- [Commonly Asked Data Structure Interview Questions on Hash Data Structure](#)
- [Commonly Asked Data Structure Interview Questions on Tree Data Structures](#)
- [Commonly Asked Data Structure Interview Questions on Graph](#)

## Commonly Asked Data Structure Interview Questions on Array

### Question 1: What is an array?

Answer: An array is a data structure consisting of a collection of elements, each identified by at least one array index or key.

## Question 2: Can an array be resized at runtime?

**Answer:** In some programming languages, arrays can be resized dynamically, while in others, such as C, the size is fixed.

## Question 3: What is the time complexity for accessing an element in an array?

**Answer:** The time complexity for accessing an element in an array is **O(1)**, as it can be accessed directly using its index.

## Question 4: What is the difference between an array and a linked list?

**Answer:** An array is a static data structure, while a linked list is a dynamic data structure. Arrays have a fixed size, and elements are

stored consecutively in memory, while linked lists can grow and do not require contiguous memory allocation.

### Question 5: How would you find the smallest and largest element in an array?

**Answer:** To find the smallest and largest elements in an array, one common approach is to iterate through the array and keep track of the smallest and largest elements encountered so far.

### Question 6: Explain the concept of a multi-dimensional array.

**Answer:** A multi-dimensional array is an array that contains other arrays. For example, a 2D array is an array of arrays, representing a matrix.

### Question 7: What is an array index out of bounds exception?

**Answer:** This error occurs when an attempt is made to access an element at an index that is outside the bounds of the array (e.g., negative index or greater than the array size).

### Question 8: How would you reverse an array in-place in linear time and constant space?

**Answer:** One approach is to use two pointers starting from the beginning and end of the array and swap the elements until they meet in the middle.

### Question 9: Explain the concept of a jagged array.

**Answer:** A jagged array is an array of arrays, where each sub-array could be of a different length.

### Question 10: How can you find duplicate elements in an array?

**Answer:** One way to find duplicate elements in an array is to use a hash set or to sort the array and then iterate through it to find consecutive duplicates.

**Question 11:** Discuss the advantages and disadvantages of using arrays.

**Answer:**

- **Advantages:** Constant time access, simple implementation, and efficient storage for contiguous data.
- **Disadvantages:** Fixed size, no support for dynamic growth, inefficient for insertions and deletions.

**Question 12:** Explain the concept of a sparse array.

**Answer:** A sparse array is an array in which most of the elements have the same value. It can be represented using a data structure that only stores the non-default (non-zero) values.

**Question 13:** What is the difference between an array and a list?

**Answer:** An array is a static data structure with a fixed size, while a list is a dynamic data structure that can grow and shrink during runtime.

## Commonly Asked Data Structure Interview Questions on Linked List

**Question 1:** What is a linked list?

**Answer:** A linked list is a linear data structure consisting of a sequence of elements, where each element points to the next one, forming a chain.

**Question 2:** What are the different types of linked lists?

**Answer:** Singly linked list, doubly linked list, and circular linked list.

### Question 3: What are the advantage of Linked List?

**Answer: Advantages of Linked Lists:**

- Dynamic memory allocation
- Efficient insertion and deletion
- Can represent complex data structures
- Can be used to implement queues and stacks
- Can be used for memory management and caching
- Can be used for garbage collection

### Question 4: What are the disadvantage of Linked List?

**Answer: Disadvantages of Linked Lists:**

- Slow random access
- More memory overhead
- Difficult to debug
- Not cache-friendly
- Can suffer from memory leaks

### Question 5: What is a cycle/loop in Singly Linked List:

**Answer:** A cycle, also known as a loop, in a singly-linked list occurs when a node in the list points back to a previous node, creating a circular path. This means that if you start traversing the list from any node, you will eventually come back to the same node, forming an infinite loop.

### Question 6: What is time complexity of Linked List operations?

**Answer:** The time complexity of common operations on a singly-linked list are as follows:

**Insertion:**

- At the beginning: O(1)

- At the end:  $O(n)$
- At a specific position:  $O(n)$

### Deletion:

- At the beginning:  $O(1)$
- At the end:  $O(n)$
- At a specific position:  $O(n)$

Search:  $O(n)$

Traversal:  $O(n)$

## Question 7: How would you compare Dynamic Arrays Vs Linked Lists?

### Answer: Dynamic Array Advantages:

- Fast random access ( $O(1)$ )
- Efficient for large data sets
- Contiguous memory allocation

### Dynamic Array Disadvantages:

- Slow insertion and deletion in the middle ( $O(n)$ )
- Fixed size, can lead to memory waste or reallocation

### Linked Lists Advantages:

- Efficient insertion and deletion in the middle ( $O(1)$ )
- Can grow and shrink dynamically
- Can represent complex data structures

### Linked Lists Disadvantages:

- Slow random access ( $O(n)$ )
- More memory overhead due to pointers
- Not cache-friendly

Dynamic arrays are more efficient for random access and large data sets, while linked lists are more efficient for operations that involve insertion and deletion in the middle. Linked lists are also more flexible and can represent complex data structures.

## Commonly Asked Data Structure Interview Questions on Stack:

### Question 1: What is a stack?

**Answer:** A stack is a linear data structure that follows the **Last-In-First-Out (LIFO)** principle.

### Question 2: What are the operations performed on a stack?

**Answer:** The common operations on a stack are push (insert an element), pop (remove the top element), and peek (view the top element).

### Question 3: How is a stack implemented in an array?

**Answer:** A stack can be implemented using an array by maintaining a pointer to the top of the stack.

### Question 4: What is the time complexity of stack operations?

**Answer:** Push, pop, and peek operations have a time complexity of  $O(1)$ .

### Question 5: What are the applications of a stack?

**Answer:** Stacks are used in various applications, such as function calls, recursion, expression evaluation, and parsing.

### Question 6: What is a stack overflow?

**Answer:** A stack overflow occurs when the stack exceeds its allocated memory.

### **Question 7: What is a stack underflow?**

**Answer:** A stack underflow occurs when the stack is empty and an attempt is made to pop an element.

### **Question 8: What is a postfix expression?**

**Answer:** A postfix expression is an expression where the operator follows the operands.

### **Question 9: How can a stack be used to evaluate a postfix expression?**

**Answer:** By pushing operands onto the stack and performing operations when operators are encountered.

### **Question 10: What is a prefix expression?**

**Answer:** A prefix expression is an expression where the operator precedes the operands.

### **Question 11: How can a stack be used to evaluate a prefix expression?**

**Answer:** By pushing operators onto the stack and performing operations when operands are encountered.

### **Question 12: How can a stack be used to check if a parenthesis expression is balanced?**

**Answer:** A stack can be used to check if a parenthesis expression is balanced by following these steps:

- Push the opening parenthesis onto the stack.

- When an closing parenthesis is encountered, pop the top element from the stack and check if it matches the closing parenthesis.
- If the stack is empty at the end of the expression, then the expression is balanced.
- If the stack is not empty, then the expression is not balanced.

## Commonly Asked Data Structure Interview Questions on Queue

### Question 1: What is a Queue?

Answer: A queue is a linear data structure that follows the **First-In-First-Out (FIFO)** principle, where elements are added at the rear (enqueue) and removed from the front (dequeue).

### Question 2: What are the different types of Queues?

Answer:

- Simple Queue
- Circular Queue
- Priority Queue
- Double-Ended Queue (Deque).

### Question 3: How is a Queue implemented in an array?

Answer: An array can be used to implement a simple queue by maintaining two pointers: front and rear. Front points to the first element, and rear points to the next available position.

### Question 4: How is a Queue implemented in a linked list?

Answer: A linked list can be used to implement a queue by creating a node for each element and maintaining a head and tail pointer. Enqueueing adds a node to the tail, and dequeuing removes a node from the head.

## Question 5: What is the time complexity of enqueue and dequeue operations in a Queue?

Answer:

- Enqueue: O(1)
- Dequeue: O(1) for simple and circular queues, O(n) for priority queues

## Question 6: What is the difference between a Queue and a Stack?

Answer: A queue follows **FIFO**, while a stack follows **Last-In-First-Out (LIFO)**.

## Question 7: What are the applications of Queues?

Answer:

- Task scheduling
- Message passing
- Simulation of real-world scenarios

## Question 8: How do you handle overflow and underflow conditions in a Queue?

Answer:

- **Overflow:** When the queue is full, throw an exception or return an error code.
- **Underflow:** When the queue is empty, throw an exception or return a null value.

## Question 9: What is a circular queue?

Answer: A circular queue is a variation of a simple queue where the rear pointer wraps around to the beginning of the array after reaching the end.

## Question 10: What is a priority queue?

**Answer:** A priority queue is a queue where elements are assigned priorities and are dequeued based on their priorities.

### Question 11: How is a priority queue implemented?

**Answer:** Priority queues can be implemented using a binary heap or a self-balancing binary search tree.

### Question 12: What is a double-ended queue (Deque)?

**Answer:** A deque is a queue that allows insertions and deletions from both ends.

### Question 13: How is a deque implemented?

**Answer:** A deque can be implemented using two stacks or a circular buffer.

### Question 14: What are the advantages of using a Queue?

**Answer:**

- Simple and efficient FIFO implementation
- Easy to enqueue and dequeue elements
- Supports multiple producers and consumers

### Question 15: What are the disadvantages of using a Queue?

**Answer:**

- Limited access to elements (only from the front or rear)
- Can be inefficient if elements need to be accessed in a non-sequential order

## Commonly Asked Data Structure Interview Questions on Heap Data Structure

### Question 1: What is a heap data structure?

**Answer:** A heap is a complete binary tree that satisfies the heap property: each node's value is greater than or equal to its children's values.

### Question 2: What are the two types of heaps?

**Answer:** Max-heap and min-heap. In a max-heap, the root node has the maximum value, while in a min-heap, the root node has the minimum value.

### Question 3: What is the time complexity of inserting an element into a heap?

**Answer:**  $O(\log n)$ , where  $n$  is the number of elements in the heap.

### Question 4: What is the time complexity of deleting an element from a heap?

**Answer:**  $O(\log n)$ , where  $n$  is the number of elements in the heap.

### Question 5: What is the time complexity of finding the minimum or maximum element in a heap?

**Answer:**  $O(1)$ , as the root node always contains the minimum or maximum element.

### Question 6: What are the applications of heaps?

**Answer:** Heap applications:

- Priority queues
- Sorting
- Finding the median
- Implementing Dijkstra's algorithm
- Network routing
- Huffman coding

## Question 7: What is the difference between a heap and a binary search tree (BST)?

**Answer:** A heap is a complete binary tree that satisfies the heap property, while a BST is a partially ordered binary tree that satisfies the BST property.

## Question 8: How do you convert a BST into a heap?

**Answer:** By performing an in-order traversal of the BST and inserting the elements into a heap.

## Question 9: How do you merge two heaps?

**Answer:** By creating a new heap and inserting the elements from both heaps into the new heap while maintaining the heap property.

## Question 10: What is the difference between a heap and a priority queue?

**Answer:** A heap is a data structure, while a priority queue is an abstract data type that can be implemented using a heap.

## Question 11: What are the advantages of using a heap?

**Answer:** Advantages of using a heap:

- Efficient insertion and extraction ( $O(\log n)$ )
- Can be used to implement priority queues
- Can be used for sorting ( $O(n \log n)$ )
- Useful for other applications, such as finding the median and implementing [Dijkstra's algorithm](#)

# Commonly Asked Data Structure Interview Questions on Hash Data Structure

## Question 1: What is a hash data structure?

**Answer:** A hash data structure is a data structure that stores key-value pairs, where the keys are hashed to determine the location of the value in the data structure.

### Question 2: What is a hash table?

**Answer:** A hash table is a data structure that implements an associative array, allowing fast retrieval of values based on unique keys. It uses a hash function to map keys to indices in an array, providing constant-time average access ( $O(1)$ ) if collisions are minimized.

### Question 3: What is a hash function?

**Answer:** A hash function is a function that takes an input of any size and produces an output of a fixed size. The output is called a **hash value or hash code**.

### Question 4: Explain how a hash function works.

**Answer:** A hash function takes an input key and maps it to a fixed-size index (hash value) within the hash table's array. Ideally, the function distributes keys evenly across the array to minimize collisions. Common hash functions include **modulo division**, **bitwise operations**, and **polynomial hashing**.

### Question 5: What is a collision?

**Answer:** A collision occurs when two different keys hash to the same value.

### Question 6: Describe different collision resolution techniques.

**Answer:**

- **Open addressing:** Use probing techniques (linear, quadratic, double hashing) to find the next available slot when a collision occurs.

- **Separate chaining:** Store key-value pairs in linked lists at each index, leading to better performance for larger data sets.

## Question 7: How are collisions handled in a hash data structure?

Answer: Collisions can be handled using various techniques, such as **chaining**, **open addressing**, and **cuckoo hashing**.

## Question 8: What is chaining?

Answer: **Chaining** is a collision resolution technique where colliding keys are stored in a linked list at the same hash value.

## Question 9: What is open addressing?

Answer: **Open addressing** is a collision resolution technique where colliding keys are stored in the same hash table, but at different locations.

## Question 10: What is separate chaining?

Answer: **Separate chaining** is a collision resolution technique used in hash tables. When two or more keys hash to the same index in the hash table, instead of overwriting the existing data, separate chaining stores each key-value pair in a linked list at that index. This allows for efficient retrieval of all elements that hash to the same index, even if there are collisions.

## Question 11: What are the trade-offs between open addressing and separate chaining?

Answer:

- **Open addressing:** Less memory overhead, but search performance degrades with collisions.
- **Separate chaining:** More memory usage, but faster search performance even with collisions.

## Question 12: What is cuckoo hashing?

**Answer:** Cuckoo hashing is a collision resolution technique that uses two hash functions to store keys in a hash table.

## Question 13: What is the load factor of a hash table?

**Answer:** The load factor of a hash table is the ratio of the number of keys stored in the table to the size of the table.

## Question 14: What is the optimal load factor for a hash table?

**Answer:** The optimal load factor for a hash table depends on the collision resolution technique used, but it is typically around 0.7.

## Question 15: Explain the concept of load factor and its impact on performance.

**Answer:** Load factor (number of elements / size of hash table) measures how full the table is. Higher load factors increase collision frequency and impact performance. Optimal values vary based on implementation and trade-offs.

## Question 16: What are the advantages of using a hash data structure?

**Answer:** Hash data structures offer fast lookup, insertion, and deletion operations. They are also space-efficient and can handle large datasets.

## Question 17: What are the disadvantages of using a hash data structure?

**Answer:** Hash data structures can suffer from collisions, which can slow down lookup operations. They also require a hash function that is both efficient and effective.

## Question 18: Explain bloom filters and their applications.

**Answer:** Bloom filters use multiple hash functions to probabilistically represent set membership, offering efficient space-time trade-offs for membership queries but not supporting direct value retrieval. Used in caching, [network security](#), and other applications.

# Commonly Asked Data Structure Interview Questions on Tree Data Structures:

## Question 1: What is a Tree?

**Answer:** A [tree](#) is a non-linear data structure consisting of nodes connected by edges. Each node contains data and [references](#) to its child nodes. It has one special node called the root, with no parent, and leaf nodes with no children.

## Question 2: Explain different types of trees.

**Answer:**

- [Binary Tree](#): Each node has at most two children (left and right).
- [Full Binary Tree](#): Every node except leaves has two children.
- [Complete Binary Tree](#): All levels are filled except possibly the last, and nodes are filled left to right.
- [Perfect Binary Tree](#): Every node has two children, and all leaves are at the same level.
- [AVL Tree](#): Self-balancing binary search tree with a height difference of at most 1 between subtrees.
- [Red-Black Tree](#): Self-balancing binary search tree with specific coloring rules to maintain balance.
- [B-Tree](#): Generalization of a binary search tree with more than two children per node.

## Question 3: What are the basic operations performed on a tree?

**Answer:**

- **Insertion:** Add a new node to the tree while maintaining its properties (e.g., ordering in search trees).
- **Deletion:** Remove a node from the tree while preserving its structure.
- **Traversal:** Visit each node in the tree exactly once in a specific order (preorder, inorder, postorder).
- **Searching:** Find a specific node with a given value based on search criteria.

**Question 4: What are the different ways to represent a tree in memory?**

**Answer:**

- **Node-based representation:** Each node stores its data and references to child nodes.
- **Array-based representation:** Use an array to store node data with calculations to find child nodes based on their positions.

**Question 5: What are the advantages and disadvantages of using trees?**

**Advantages:**

- Efficient for hierarchical data representation and organization.
- Fast searching and traversal in balanced trees.

**Disadvantages:**

- Memory overhead due to storing pointers or references.
- Not efficient for storing large amounts of unstructured data.

**Question 6: When would you choose a tree over other data structures like arrays or linked lists?**

**Answer:** Trees are ideal for hierarchical data, maintaining relationships between elements, and efficient searching based on order. Arrays or

linked lists are better for simple linear data or frequent insertions/deletions at specific positions.

### Question 7: Explain the concept of a binary search tree.

**Answer:** A binary search tree has a specific ordering property: the data in the left subtree is less than the root, and the data in the right subtree is greater than the root. This allows for efficient searching by comparing values with the root and navigating left or right accordingly.

### Question 8: How do self-balancing trees like AVL or Red-Black trees work?

**Answer:** These trees automatically adjust their structure after insertions or deletions to maintain a balanced height, ensuring efficient search and insertion/deletion operations. They achieve this through specific rules and rotations based on node heights and colors.

### Question 9: Describe the different tree traversal methods (preorder, inorder, postorder).

**Answer:**

- Preorder: Visit root, then left subtree, then right subtree.
- Inorder: Visit left subtree, then root, then right subtree.
- Postorder: Visit left subtree, then right subtree, then root.

Each traversal method has different purposes. Inorder is useful for printing sorted elements in a binary search tree, while preorder might be used for copying tree structure.

### Question 10: How can you convert a binary search tree into a sorted array?

**Answer:** One efficient way is to use an inorder traversal of the tree. Since the tree is sorted, visiting nodes in this order will result in a sorted array.

## Question 11: Explain the concept of a minimum spanning tree.

**Answer:** A minimum spanning tree is a subgraph of a connected, undirected graph that includes all vertices but with the minimum total edge weight, connecting all nodes without cycles. It has applications in network routing and clustering algorithms.

## Question 12: Describe the use of trees in real-world scenarios.

**Answer:** Trees are used in various domains, including:

- File systems (directory structure)
- XML or JSON data representation
- Decision trees for machine learning
- Game AI (representing game states and possible actions)
- Social networks (representing user relationships)

# Commonly Asked Data Structure Interview Questions on Graph:

## Question 1: What is a graph?

**Answer:** A graph is a data structure consisting of a set of vertices (nodes) and a set of edges that connect pairs of vertices. Graphs are used to represent relationships between objects, such as social networks, road networks, and computer networks.

## Question 2: Explain common graph representations.

**Answer:**

- **Adjacency matrix:** A 2D array where rows and columns represent nodes, and values indicate the existence of an edge between them. Efficient for space usage, but can be slow for sparse graphs.
- **Adjacency list:** An array of linked lists or other data structures, where each list stores nodes connected to a specific

node. Efficient for sparse graphs and adjacency queries, but may require more space.

### Question 3: Differentiate between directed and undirected graphs.

Answer:

- **Directed graphs:** Edges have a direction, signifying one-way relationships.
- **Undirected graphs:** Edges have no direction, representing bidirectional relationships.

### Question 4: Describe common graph types.

Answer:

- **Simple graphs:** Undirected, no loops or multiple edges between nodes.
- **Complete graphs:** Every node is connected to every other node.
- **Trees:** No cycles, a single root node connects to child nodes that don't form cycles.
- **Bipartite graphs:** Can be divided into two disjoint sets where only nodes in different sets connect.

### Question 5: Discuss time and space complexity of basic graph operations.

Answer:

- **Traversal (DFS, BFS):**  $O(V + E)$  for both time and space, where  $V$  is the number of nodes and  $E$  is the number of edges.
- **Insertion:**  $O(1)$  for constant-time operations in both adjacency list and matrix, although insertion into **sparse matrices** requiring reallocation might have amortized cost.
- **Deletion:**  $O(\text{degree of the node})$  for adjacency lists,  $O(V^2)$  for adjacency matrices due to potential row/column shifts.

### Question 6: Explain Dijkstra's algorithm and its applications.

**Answer:** This algorithm finds the shortest path between two nodes in a weighted graph. It is used in route planning, network optimization, and other problems involving finding minimum-cost paths.

**Question 7: Compare DFS and BFS algorithms: strengths, weaknesses, and use cases.**

**Answer:**

- **DFS:** Explores deeply before exploring breadth-wise, efficient for finding connected components, good for detecting cycles.
- **BFS:** Explores breadth-wise, efficient for finding shortest paths in unweighted graphs, useful for level-order traversals.

**Question 8: Describe topological sorting: algorithm and applications.**

**Answer:**

- **Topological sorting:** Orders nodes such that edges always point from earlier nodes to later ones, required for tasks with dependencies.
- **Application:** Used in job scheduling, software dependency management, and circuit design.

**Question 9: Explain minimum spanning trees: algorithms and their significance.**

**Answer:**

- **Minimum spanning trees:** Finds a subset of edges that connects all nodes with minimum total weight while avoiding cycles.
- **Significance:** Used in network communication, clustering.

**Related posts:**

- [Last Minute Notes – DS](#)
- [Commonly Asked Data Structure Interview Questions](#)

- [Commonly Asked Algorithm Interview Questions](#)

Some other important Tutorials:

- [DSA Tutorial](#)
- [System Design Tutorial](#)
- [Software Development Roadmap](#)
- [Roadmap to become a Product Manager](#)
- [Learn SAP](#)
- [Learn SEO](#)

Unlock the Power of Placement Preparation!

Feeling lost in OS, DBMS, CN, SQL, and DSA chaos? Our [Complete Interview Preparation](#) Course is the ultimate guide to conquer placements. Trusted by over 100,000+ geeks, this course is your roadmap to interview triumph.

Ready to dive in? Explore our Free Demo Content and join our [Complete Interview Preparation](#) course.

Recommended Problems

[\*\*Frequently asked DSA Problems\*\*](#)

Solve Problems

203

[Suggest improvement](#)

Next

[\*\*Data Analyst Interview Questions  
and Answers\*\*](#)

[\*\*Share your thoughts in the comments\*\*](#)

[\*\*Add Your Comment\*\*](#)

## Similar Reads

[Commonly asked JavaScript Interview Questions | Set 1](#)

[Commonly Asked C Programming Interview Questions | Set 1](#)

[Commonly Asked C Programming Interview Questions | Set 2](#)

[Commonly Asked C++ Interview Questions | Set 1](#)

[Commonly Asked Java Programming Interview Questions | Set 2](#)

[Most Commonly Asked System Design Interview Problems/Questions](#)

[Commonly Asked Algorithm Interview Questions](#)

[Commonly Asked C++ Interview Questions | Set 2](#)

[Commonly Asked Questions in Goldman Sachs Interviews](#)

[Top 50 Problems on Heap Data Structure asked in SDE Interviews](#)



GeeksforGeeks

Article Tags : [interview-preparation](#), [placement preparation](#), [DSA](#)



A-143, 9th Floor, Sovereign Corporate  
Tower, Sector-136, Noida, Uttar Pradesh -  
201305



## Company

[About Us](#)

[Legal](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

[Advertise with us](#)

[GFG Corporate Solution](#)

[Placement Training Program](#)

## Explore

[Job-A-Thon Hiring Challenge](#)

[Hack-A-Thon](#)

[GfG Weekly Contest](#)

[Offline Classes \(Delhi/NCR\)](#)

[DSA in JAVA/C++](#)

[Master System Design](#)

[Master CP](#)

[GeeksforGeeks Videos](#)

[Geeks Community](#)

## Languages

[Python](#)

[Java](#)

[C++](#)

[PHP](#)

[GoLang](#)

[SQL](#)

[R Language](#)

[Android Tutorial](#)

## DSA

[Data Structures](#)

[Algorithms](#)

[DSA for Beginners](#)

[Basic DSA Problems](#)

[DSA Roadmap](#)

[DSA Interview Questions](#)

[Competitive Programming](#)

## Data Science & ML

[Data Science With Python](#)

[Data Science For Beginner](#)

[Machine Learning Tutorial](#)

[ML Maths](#)

[Data Visualisation Tutorial](#)

[Pandas Tutorial](#)

## Web Technologies

[HTML](#)

[CSS](#)

[JavaScript](#)

[TypeScript](#)

[ReactJS](#)

[NextJS](#)

[NumPy Tutorial](#)[NodeJS](#)[NLP Tutorial](#)[Bootstrap](#)[Deep Learning Tutorial](#)[Tailwind CSS](#)

## Python Tutorial

[Python Programming Examples](#)[Django Tutorial](#)

## Computer Science

[GATE CS Notes](#)[Python Projects](#)[Operating Systems](#)[Python Tkinter](#)[Computer Network](#)[Web Scraping](#)[Database Management System](#)[OpenCV Tutorial](#)[Software Engineering](#)[Python Interview Question](#)[Digital Logic Design](#)[Engineering Maths](#)

## DevOps

[Git](#)

## System Design

[AWS](#)[High Level Design](#)[Docker](#)[Low Level Design](#)[Kubernetes](#)[UML Diagrams](#)[Azure](#)[Interview Guide](#)[GCP](#)[Design Patterns](#)[DevOps Roadmap](#)[OOAD](#)[System Design Bootcamp](#)[Interview Questions](#)

## School Subjects

[Mathematics](#)

## Commerce

[Physics](#)[Accountancy](#)[Chemistry](#)[Business Studies](#)[Biology](#)[Economics](#)[Social Science](#)[Management](#)[English Grammar](#)[HR Management](#)[Finance](#)[Income Tax](#)

**UPSC Study Material**

- [Polity Notes](#)
- [Geography Notes](#)
- [History Notes](#)
- [Science and Technology Notes](#)
- [Economy Notes](#)
- [Ethics Notes](#)
- [Previous Year Papers](#)

**Preparation Corner**

- [Company-Wise Recruitment Process](#)
- [Resume Templates](#)
- [Aptitude Preparation](#)
- [Puzzles](#)
- [Company-Wise Preparation](#)
- [Companies](#)
- [Colleges](#)

**Competitive Exams**

- [JEE Advanced](#)
- [UGC NET](#)
- [SSC CGL](#)
- [SBI PO](#)
- [SBI Clerk](#)
- [IBPS PO](#)
- [IBPS Clerk](#)

**More Tutorials**

- [Software Development](#)
- [Software Testing](#)
- [Product Management](#)
- [Project Management](#)
- [Linux](#)
- [Excel](#)
- [All Cheat Sheets](#)

**Free Online Tools**

- [Typing Test](#)
- [Image Editor](#)
- [Code Formatters](#)
- [Code Converters](#)
- [Currency Converter](#)
- [Random Number Generator](#)
- [Random Password Generator](#)

**Write & Earn**

- [Write an Article](#)
- [Improve an Article](#)
- [Pick Topics to Write](#)
- [Share your Experiences](#)
- [Internships](#)

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved



# Java Interview Questions and Answers

Last Updated : 03 Apr, 2024

Java is one of the most popular programming languages in the world, known for its versatility, portability, and wide range of applications. Java is the most used language in top companies such as Uber, Airbnb, Google, Netflix, Instagram, Spotify, Amazon, and many more because of its features and performance.

In this article, we will provide **200+ Core Java Interview Questions** tailored for both freshers and experienced professionals with 3, 5, and 8 years of experience. Here, we cover everything, including core Java concepts, Object-Oriented Programming (OOP), multithreading, exception handling, design patterns, Java Collections, and more, that will surely help you to crack Java interviews.



## Table of Content

- [Java Interview questions for Freshers](#)
- [Java Intermediate Interview Questions](#)
- [Java Interview Questions For Experienced](#)
- [Java Difference Interview Questions](#)

## Java Interview questions for Freshers

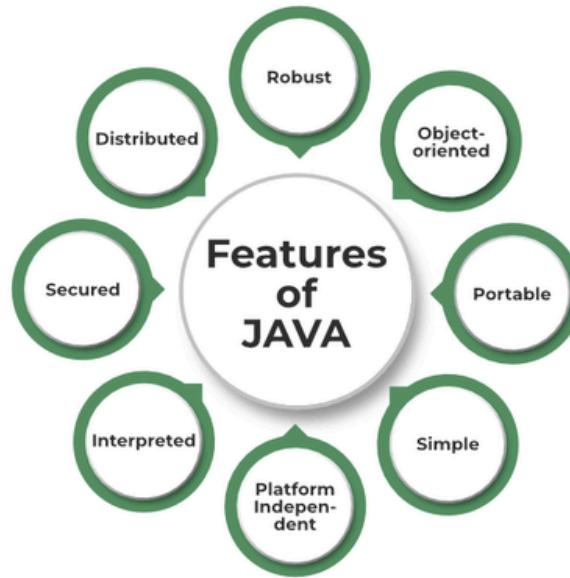
### 1. Is Java Platform Independent if then how?

Yes, Java is a Platform Independent language. Unlike many programming languages javac compiler compiles the program to form a bytecode or .class file. This file is independent of the software or hardware running but needs a JVM(Java Virtual Machine) file preinstalled in the operating system for further execution of the bytecode.

Although **JVM is platform dependent**, the bytecode can be created on any System and can be executed in any other system despite hardware or software being used which makes Java platform independent.

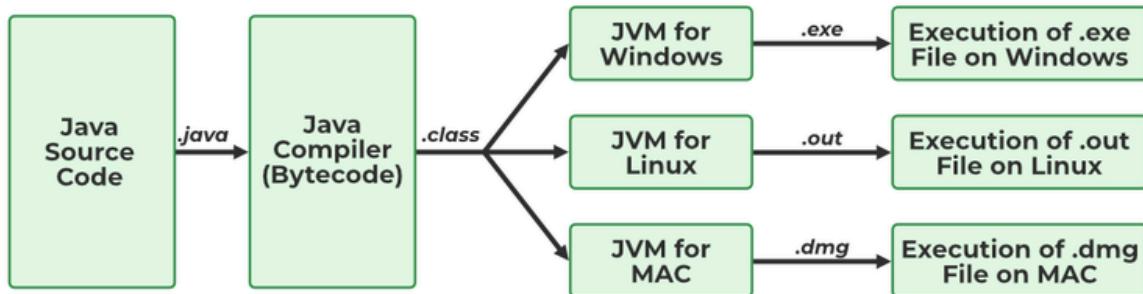
### 2. What are the top Java Features?

Java is one the most famous and most used language in the real world, there are many features in Java that makes it better than any other language some of them are mentioned below:



- **Simple:** Java is quite simple to understand and the syntax
- **Platform Independent:** Java is platform independent means we can run the same program in any software and hardware and will get the same result.
- **Interpreted:** Java is interpreted as well as a compiler-based language.
- **Robust:** features like Garbage collection, exception handling, etc that make the language robust.
- **Object-Oriented:** Java is an object-oriented language that supports the concepts of class, objects, four pillars of OOPS, etc.
- **Secured:** As we can directly share an application with the user without sharing the actual program makes Java a secure language.
- **High Performance:** faster than other traditional interpreted programming languages.
- **Dynamic:** supports dynamic loading of classes and interfaces.
- **Distributed:** feature of Java makes us able to access files by calling the methods from any machine connected.
- **Multithreaded:** deal with multiple tasks at once by defining multiple threads
- **Architecture Neutral:** it is not dependent on the architecture.

### 3. What is JVM?

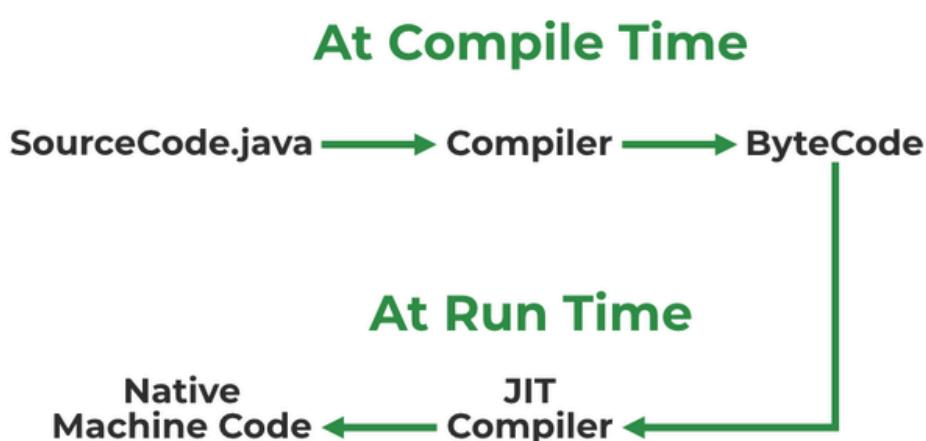


JVM stands for Java Virtual Machine it is a Java interpreter. It is responsible for loading, verifying, and executing the bytecode created in Java.

Although it is platform dependent which means the software of JVM is different for different Operating Systems it plays a vital role in making Java platform Independent.

To know more about the topic refer to [JVM in Java](#).

### 4. What is JIT?

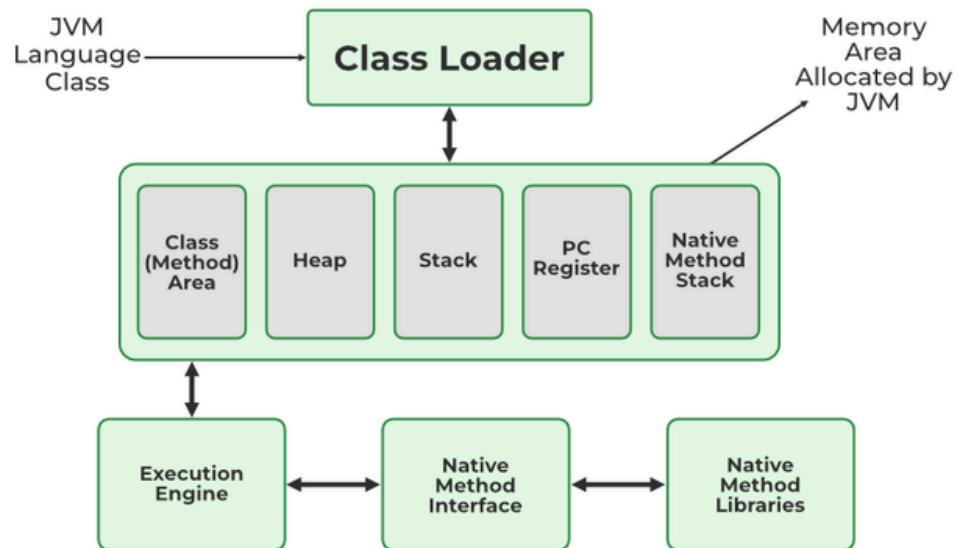


JIT stands for (Just-in-Time) compiler is a part of JRE (Java Runtime Environment), it is used for better performance of the Java applications during run-time. The use of JIT is mentioned in step by step process mentioned below:

1. Source code is compiled with **javac** compiler to form bytecode
2. Bytecode is further passed on to JVM
3. JIT is a part of JVM, JIT is responsible for compiling bytecode into native machine code at run time.
4. The JIT compiler is enabled throughout, while it gets activated when a method is invoked. For a compiled method, the JVM directly calls the compiled code, instead of interpreting it.
5. As JVM calls the compiled code that increases the performance and speed of the execution.

To know more about the topic refer to [JIT in Java](#).

## 5. What are Memory storages available with JVM?



JVM consists of a few memory storages as mentioned below:

1. Class(Method) Area: stores class-level data of every class such as the runtime constant pool, field, and method data, and the code for methods.
2. Heap: Objects are created or objects are stored. It is used to allocate memory to objects during run time.

3. Stack: stores data and partial results which will be needed while returning value for method and performing dynamic linking
4. Program Counter Register: stores the address of the Java virtual machine instruction currently being executed.
5. Native Method Stack: stores all the native methods used in the application.

To know more about the topic refer to [JVM Memory Storages](#).

## 6. What is a classloader?

Classloader is the part of JRE(Java Runtime Environment), during the execution of the bytecode or created .class file classloader is responsible for dynamically loading the java classes and interfaces to JVM(Java Virtual Machine). Because of classloaders Java run time system does not need to know about files and file systems.

To know more about the topic refer to [ClassLoader in Java](#).

## 7. Difference between JVM, JRE, and JDK.

**JVM:** JVM also known as Java Virtual Machine is a part of JRE. JVM is a type of interpreter responsible for converting bytecode into machine-readable code. JVM itself is platform dependent but it interprets the bytecode which is the platform-independent reason why Java is platform-independent.

**JRE:** JRE stands for Java Runtime Environment, it is an installation package that provides an environment to run the Java program or application on any machine.

**JDK:** JDK stands for Java Development Kit which provides the environment to develop and execute Java programs. JDK is a package that includes two things Development Tools to provide an environment to develop your Java programs and, JRE to execute Java programs or applications.

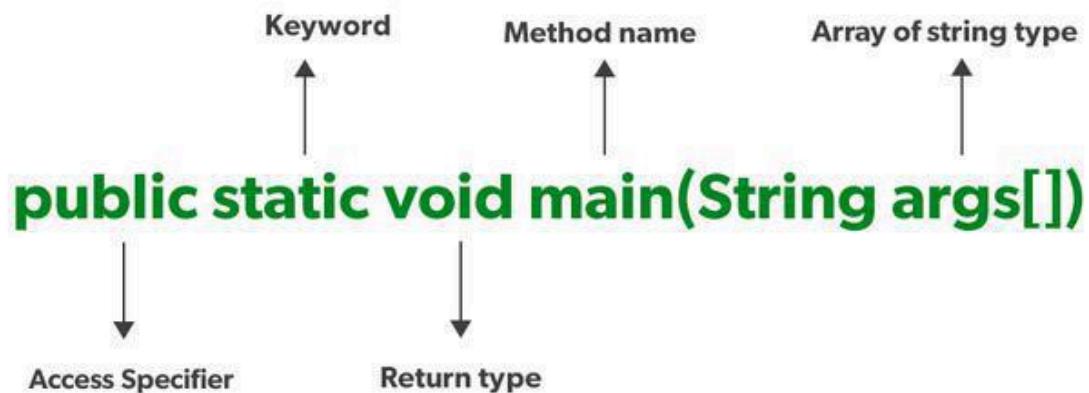
To know more about the topic refer to the [Differences between JVM, JRE, and JDK](#).

## 8. What are the differences between Java and C++?

Basis	C++	Java
Platform	C++ is Platform Dependent	Java is Platform Independent
Application	C++ is mainly used for System Programming	Java is Mainly used for Application Programming
Hardware	C++ is nearer to hardware	Java is not so interactive with hardware
Global Scope	C++ supports global and namespace scope.	Java doesn't support global scope.
Not Supporting	Functionality supported in Java but not in C++ are: <ul style="list-style-type: none"> <li>• thread support</li> <li>• documentation comment</li> <li>• unsigned right shift(&gt;&gt;&gt;)</li> </ul>	Functionality supported in C++ but not in Java are: <ul style="list-style-type: none"> <li>• goto</li> <li>• Pointers</li> <li>• Call by reference</li> <li>• Structures and Unions</li> <li>• Multiple Inheritance</li> <li>• Virtual Functions</li> </ul>
OOPS	C++ is an object-oriented language. It is	Java is also an object-oriented language. It is a single root hierarchy

Basis	C++	Java
	not a single root hierarchy .	as everything gets derived from a single class (java.lang.Object).
Inheritance Tree	C++ always creates a new inheritance tree.	Java uses a Single inheritance tree as classes in Java are the child of object classes in Java.

## 9. Explain public static void main(String args[]) in Java.



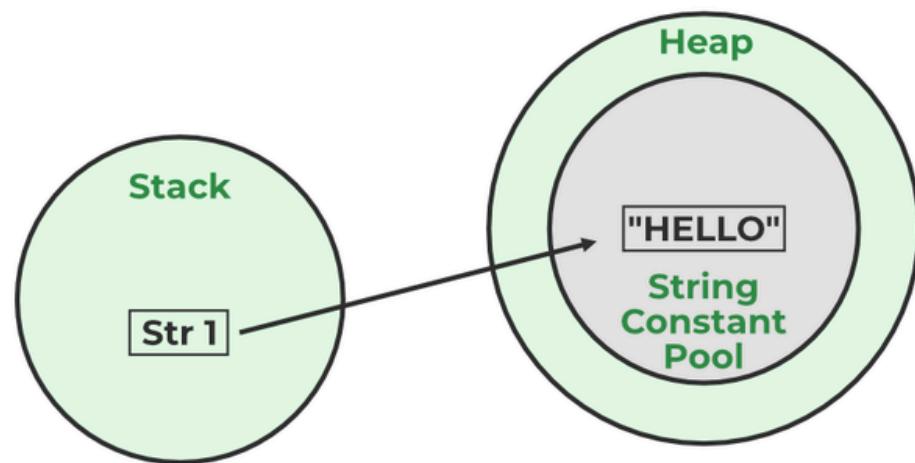
Unlike any other programming language like C, C++, etc. In Java, we declared the main function as a public static void main (String args[]). The meanings of the terms are mentioned below:

- 1. public**: the public is the access modifier responsible for mentioning who can access the element or the method and what is the limit. It is responsible for making the main function globally available. It is made public so that JVM can invoke it from outside the class as it is not present in the current class.

2. **static:** static is a keyword used so that we can use the element without initiating the class so to avoid the unnecessary allocation of the memory.
3. **void:** void is a keyword and is used to specify that a method doesn't return anything. As the main function doesn't return anything we use void.
4. **main:** main represents that the function declared is the main function. It helps JVM to identify that the declared function is the main function.
5. **String args[]:** It stores Java command-line arguments and is an array of type java.lang.String class.

## 10. What is Java String Pool?

A Java String Pool is a place in heap memory where all the strings defined in the program are stored. A separate place in a stack is there where the variable storing the string is stored. Whenever we create a new string object, JVM checks for the presence of the object in the String pool, If String is available in the pool, the same object reference is shared with the variable, else a new object is created.



### Example:

```
String str1="Hello";
// "Hello" will be stored in String Pool
// str1 will be stored in stack memory
```

## 11. What will happen if we declare don't declare the main as static?

We can declare the main method without using static and without getting any errors. But, the main method will not be treated as the entry point to the application or the program.

## 12. What are Packages in Java?

Packages in Java can be defined as the grouping of related types of classes, interfaces, etc providing access to protection and namespace management.

## 13. Why Packages are used?

Packages are used in Java in order to prevent naming conflicts, control access, and make searching/locating and usage of classes, interfaces, etc easier.

## 14. What are the advantages of Packages in Java?

There are various advantages of defining packages in Java.

- Packages avoid name clashes.
- The Package provides easier access control.
- We can also have the hidden classes that are not visible outside and are used by the package.
- It is easier to locate the related classes.

## 15. How many types of packages are there in Java?

There are two types of packages in Java

- User-defined packages
- Build In packages

## 16. Explain different data types in Java.

There are 2 types of data types in Java as mentioned below:

1. Primitive Data Type
2. Non-Primitive Data Type or Object Data type

Primitive Data Type: Primitive data are single values with no special capabilities. There are 8 primitive data types:

- **boolean**: stores value true or false
- **byte**: stores an 8-bit signed two's complement integer
- **char**: stores a single 16-bit Unicode character
- **short**: stores a 16-bit signed two's complement integer
- **int**: stores a 32-bit signed two's complement integer
- **long**: stores a 64-bit two's complement integer
- **float**: stores a single-precision 32-bit IEEE 754 floating-point
- **double**: stores a double-precision 64-bit IEEE 754 floating-point

Non-Primitive Data Type: Reference Data types will contain a memory address of the variable's values because it is not able to directly store the values in the memory. Types of Non-Primitive are mentioned below:

- Strings
- Array
- Class
- Object
- Interface

## 17. When a byte datatype is used?

A byte is an 8-bit signed two-complement integer. The minimum value supported by bytes is -128 and 127 is the maximum value. It is used in conditions where we need to save memory and the limit of numbers needed is between -128 to 127.

## 18. Can we declare Pointer in Java?

No, Java doesn't provide the support of Pointer. As Java needed to be more secure because which feature of the pointer is not provided in Java.

## 19. What is the default value of byte datatype in Java?

The default value of the byte datatype in Java is 0.

## 20. What is the default value of float and double datatype in Java?

The default value of the float is 0.0f and of double is 0.0d in Java.

## 21. What is the Wrapper class in Java?

Wrapper, in general, is referred to a larger entity that encapsulates a smaller entity. Here in Java, the wrapper class is an object class that encapsulates the primitive data types.

The primitive data types are the ones from which further data types could be created. For example, integers can further lead to the construction of long, byte, short, etc. On the other hand, the string cannot, hence it is not primitive.

Getting back to the wrapper class, Java contains 8 wrapper classes. They are Boolean, Byte, Short, Integer, Character, Long, Float, and Double. Further, custom wrapper classes can also be created in Java which is similar to the concept of Structure in the C programming language. We create our own wrapper class with the required data types.

## 22. Why do we need wrapper classes?

The wrapper class is an object class that encapsulates the primitive data types, and we need them for the following reasons:

1. Wrapper classes are final and immutable
2. Provides methods like valueOf(), parseInt(), etc.
3. It provides the feature of autoboxing and unboxing.

### 23. Differentiate between instance and local variables.

Instance Variable	Local Variable
Declared outside the method, directly invoked by the method.	Declared within the method.
Has a default value.	No default value
It can be used throughout the class.	The scope is limited to the method.

### 24. What are the default values assigned to variables and instances in Java?

In Java When we haven't initialized the instance variables then the compiler initializes them with default values. The default values for instances and variables depend on their data types. Some common types of default data types are:

- The default value for numeric types (byte, short, int, long, float, and double) is 0.
- The default value for the boolean type is false.
- The default value for object types (classes, interfaces, and arrays) is null.
- The null character, “\u0000,” is the default value for the char type.

#### Example:

Java

```

// Java Program to demonstrate use of default values
import java.io.*;
class GFG {
    // static values
    static byte b;
    static int i;
}

```

```
static long l;
static short s;
static boolean bool;
static char c;
static String str;
static Object object;
static float f;
static double d;
static int[] Arr;
public static void main(String[] args)
{
    // byte value
    System.out.println("byte value" + b);
    // short value
    System.out.println("short value" + s);
    // int value
    System.out.println("int value" + i);
    // long value
    System.out.println("long value" + l);
    System.out.println("boolean value" + bool);
    System.out.println("char value" + c);
    System.out.println("float value" + f);
    System.out.println("double value" + d);
    System.out.println("string value" + str);
    System.out.println("object value" + object);
    System.out.println("Array value" + Arr);
}
}
```

## Output

```
byte value0
short value0
int value0
long value0
boolean valuefalse
char value
float value0.0
double value0.0
string valuenull
object valuenull
Array valuenull
```

## 25. What is a Class Variable?

In Java, a class variable (also known as a static variable) is a variable that is declared within a class but outside of any method, constructor, or block. Class variables are declared with the static keyword, and they are shared by all instances (objects) of the class as well as by the class itself. No matter how many objects are derived from a class, each class variable would only exist once.

**Example:**

Java

```
// Java program to demonstrate use of Class Variable
class GFG {
    public static int ctr = 0;
    public GFG() { ctr++; }
    public static void main(String[] args)
    {
        GFG obj1 = new GFG();
        GFG obj2 = new GFG();
        GFG obj3 = new GFG();
        System.out.println("Number of objects created are "
                           + GFG.ctr);
    }
}
```

**Output**

Number of objects created are 3

## 26. What is the default value stored in Local Variables?

There is no default value stored with local variables. Also, primitive variables and objects don't have any default values.

## 27. Explain the difference between instance variable and a class variable.

**Instance Variable:** A class variable without a static modifier known as an instance variable is typically shared by all instances of the class. These variables can have distinct values among several objects. The contents of an instance variable are completely independent of one object instance from another because they are related to a specific object instance of the class.

**Example:**

Java

```
// Java Program to demonstrate Instance Variable
import java.io.*;
class GFG {
    private String name;
    public void setName(String name) { this.name = name; }
    public String getName() { return name; }
    public static void main(String[] args)
    {
        GFG obj = new GFG();
        obj.setName("John");
        System.out.println("Name " + obj.getName());
    }
}
```

**Output**

Name John

**Class Variable:** Class Variable variable can be declared anywhere at the class level using the keyword static. These variables can only have one value when applied to various objects. These variables can be shared by all class members since they are not connected to any specific object of the class.

**Example:**

### Java

```
// Java Program to demonstrate Class Variable
import java.io.*;
class GFG {
    // class variable
    private static final double PI = 3.14159;
    private double radius;
    public GFG(double radius) { this.radius = radius; }
    public double getArea() { return PI * radius * radius; }
    public static void main(String[] args)
    {
        GFG obj = new GFG(5.0);
        System.out.println("Area of circle: "
                           + obj.getArea());
    }
}
```

**Output**

Area of circle: 78.53975

## 28. What is a static variable?

The static keyword is used to share the same variable or method of a given class. Static variables are the variables that once declared then a single copy of the variable is created and shared among all objects at the class level.

## 29. What is the difference between System.out, System.err, and System.in?

**System.out** – It is a PrintStream that is used for writing characters or can be said it can output the data we want to write on the Command Line Interface console/terminal.

**Example:**

Java

```
// Java Program to implement
// System.out
import java.io.*;
// Driver Class
class GFG {
    // Main Function
    public static void main(String[] args)
    {
        // Use of System.out
        System.out.println("");
    }
}
```

**System.err** – It is used to display error messages.

**Example:**

**Java**

```

    // Java program to demonstrate
    // System.err
    import java.io.*;
    // Driver Class
    class GFG {
        // Main function
        public static void main(String[] args)
        {
            // Printing error
            System.err.println(
                "This is how we throw error with System.err");
        }
    }

```

**Output:**

This is how we throw error with System.err

Although, System.err have many similarities both of them have quite a lot of difference also, let us check them.

System.out	System.err
It will print to the standard out of the system.	It will print to the standard error.
It is mostly used to display results on the console.	It is mostly used to output error texts.

System.out	System.err
It gives output on the console with the default(black) color.	It also gives output on the console but most of the IDEs give it a red color to differentiate.

**System.in** – It is an InputStream used to read input from the terminal Window. We can't use the System.in directly so we use Scanner class for taking input with the system.in.

**Example:**

### Java

```

    // Java Program to demonstrate
    // System.in
    import java.util.*;
    // Driver Class
    class Main {
        // Main Function
        public static void main(String[] args)
        {
            // Scanner class with System.in
            Scanner sc = new Scanner(System.in);
            // Taking input from the user
            int x = sc.nextInt();
            int y = sc.nextInt();
            // Printing the output
            System.out.printf("Addition: %d", x + y);
        }
    }

```

**Output:**

```

3
4
Addition: 7

```

**30. What do you understand by an IO stream?**



Java brings various Streams with its I/O package that helps the user to perform all the input-output operations. These streams support all types of objects, data types, characters, files, etc to fully execute the I/O operations.

### 31. What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?

The key difference between them is that byte stream data is read and written by input/output stream classes. Characters are handled by the Reader and Writer classes. In contrast to Reader/Writer classes, which accept character arrays as parameters, input/output stream class methods accept byte arrays. In comparison to input/output streams, the Reader/Writer classes are more efficient, handle all Unicode characters, and are useful for internalization. Use Reader/Writer classes instead of binary data, such as pictures, unless you do so.

**Example:**

**Java**

```


// Java Program to demonstrate Reading Writing Binary Data
// with InputStream/OutputStream
import java.io.*;

class GFG {
    public static void main(String[] args) {
        try {
    

```

```
// Writing binary data to a file using OutputStream
byte[] data = {(byte) 0xe0, 0x4f, (byte) 0xd0, 0x20,
(byte) 0xea};

OutputStream os = new FileOutputStream("data.bin");
os.write(data);
os.close();

// Reading binary data from a file using InputStream
InputStream is = new FileInputStream("data.bin");
byte[] newData = new byte[5];

is.read(newData);
is.close();

// Printing the read data
for (byte b : newData) {
    System.out.println(b);
}
} catch (IOException e) {
    e.printStackTrace();
}
}
```

## Output

-32  
79  
-48  
32  
-22

32. What are the super most classes for all the streams?

All the stream classes can be divided into two types of classes that are `ByteStream` classes and `CharacterStream` Classes. The `ByteStream` classes are further divided into `InputStream` classes and `OutputStream` classes. `CharacterStream` classes are also divided into `Reader` classes and `Writer` classes. The SuperMost classes for all the `InputStream` classes is `java.io.InputStream` and for all the output stream classes is `java.io.OutputStream`. Similarly, for all the reader classes, the super-most class is `java.io.Reader`, and for all the writer classes, it is `java.io.Writer`.

### 33. What are the FileInputStream and FileOutputStream?

To read and write data, Java offers I/O Streams. A Stream represents an input source or an output destination, which could be a file, an i/o device, another program, etc. **FileInputStream** in Java is used to read data from a file as a stream of bytes. It is mostly used for reading binary data such as images, audio files, or serialized objects.

**Example:**

```
File file = new File("path_of_the_file");
FileInputStream inputStream = new FileInputStream(file);
```

In Java, the **FileOutputStream** function is used to write data byte by byte into a given file or file descriptor. Usually, raw byte data, such as pictures, is written into a file using FileOutputStream.

**Example:**

```
File file = new File("path_of_the_file");
FileOutputStream outputStream = new FileOutputStream(file);
```

### 34. What is the purpose of using BufferedInputStream and BufferedOutputStream classes?

When we are working with the files or stream then to increase the Input/Output performance of the program we need to use the BufferedInputStream and BufferedOutputStream classes. These both classes provide the capability of buffering which means that the data will be stored in a buffer before writing to a file or reading it from a stream. It also reduces the number of times our OS needs to interact with the network or the disk. Buffering allows programs to write a big amount of data instead of writing it in small chunks. This also reduces the overhead of accessing the network or the disk.

```
BufferedInputStream(InputStream inp);
// used to create the bufferinput stream and save the arguments.

BufferedOutputStream(OutputStream output);
// used to create a new buffer with the default size.
```

### 35. What are FilterStreams?

**Stream filter or Filter Streams** returns a stream consisting of the elements of this stream that match the given predicate. While working filter() it doesn't actually perform filtering but instead creates a new stream that, when traversed, contains the elements of initial streams that match the given predicate.

#### Example:

```
FileInputStream fis =new FileInputStream("file_path");
FilterInputStream = new BufferedInputStream(fis);
```

### 36. What is an I/O filter?

An I/O filter also defined as an Input Output filter is an object that reads from one stream and writes data to input and output sources. It used java.io package to use this filter.

### 37. How many ways you can take input from the console?

There are two methods to take input from the console in Java mentioned below:

1. Using Command line argument
2. Using BufferedReader Class

### 3. Using Console Class

### 4. Using Scanner Class

The program demonstrating the use of each method is given below.

**Example:**

**Java**

```

// Java Program to implement input
// using Command Line argument
import java.io.*;
class GFG {
    public static void main(String[] args)
    {
        // check if length of args array is
        // greater than 0
        if (args.length > 0) {
            System.out.println(
                "The command line arguments are:");
            // iterating the args array and printing
            // the command line arguments
            for (String val : args)
                System.out.println(val);
        }
        else
            System.out.println("No command line "
                + "arguments found.");
    }
}
// Use below commands to run the code
// javac GFG.java
// java Main GeeksforGeeks


```

**Java**

```

// Java Program to implement
// Buffer Reader Class
import java.io.*;
class GFG {
    public static void main(String[] args)
        throws IOException
    {
        // Enter data using BufferedReader
        BufferedReader read = new BufferedReader(
            new InputStreamReader(System.in));
        // Reading data using readLine
        String x = read.readLine();
        // Printing the read line
    }
}


```

```
        System.out.println(x);
    }
}
```

## Java

```
// Java program to implement input
// Using Console Class
public class GfG {
    public static void main(String[] args)
    {
        // Using Console to input data from user
        String x = System.console().readLine();
        System.out.println("You entered string " + x);
    }
}
```

## Java

```
// Java program to demonstrate
// working of Scanner in Java
import java.util.Scanner;
class GfG {
    public static void main(String args[])
    {
        // Using Scanner for Getting Input from User
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        System.out.println("You entered string " + str);
    }
}
```

## Output:

GeeksforGeeks

## 38. Difference in the use of print, println, and printf.

print, println, and printf all are used for printing the elements but print prints all the elements and the cursor remains in the same line. println shifts the cursor to next line. And with printf we can use format identifiers too.

### 39. What are operators?

Operators are the special types of symbols used for performing some operations over variables and values.

### 40. How many types of operators are available in Java?

All types of operators in Java are mentioned below:

1. [Arithmetic Operators](#)
2. [Unary Operators](#)
3. [Assignment Operator](#)
4. [Relational Operators](#)
5. [Logical Operators](#)
6. [Ternary Operator](#)
7. [Bitwise Operators](#)
8. [Shift Operators](#)
9. [instance of operator](#)

Postfix operators are considered as the highest precedence according to Java operator precedence.

### 41. Explain the difference between >> and >>> operators.

Operators like >> and >>> seem to be the same but act a bit differently. >> operator shifts the sign bits and the >>> operator is used in shifting out the zero-filled bits.

**Example:**

Java

```
// Java Program to demonstrate
// >&gt; and >&gt;&gt; operators
import java.io.*;
// Driver
class GFG {
    public static void main(String[] args)
    {
        int a = -16, b = 1;
        // Use of >&gt;
        System.out.println(a >&gt; b);
        a = -17;
        b = 1;
        // Use of >&gt;&gt;
        System.out.println(a >&gt;&gt; b);
    }
}
```

## Output

-8  
2147483639

## 42. Which Java operator is right associative?

There is only one operator which is right associative which is = operator.

## 43. What is dot operator?

The Dot operator in Java is used to access the instance variables and methods of class objects. It is also used to access classes and sub-

packages from the package.

#### 44. What is covariant return type?

The covariant return type specifies that the return type may vary in the same direction as the subclass. It's possible to have different return types for an overriding method in the child class, but the child's return type should be a subtype of the parent's return type and because of that overriding method becomes variant with respect to the return type.

We use covariant return type because of the following reasons:

- Avoids confusing type casts present in the class hierarchy and makes the code readable, usable, and maintainable.
- Gives liberty to have more specific return types when overriding methods.
- Help in preventing run-time ClassCastException on returns.

#### 45. What is the transient keyword?

The transient keyword is used at the time of serialization if we don't want to save the value of a particular variable in a file. When JVM comes across a transient keyword, it ignores the original value of the variable and saves the default value of that variable data type.

#### 46. What's the difference between the methods sleep() and wait()?

Sleep()	Wait()
The sleep() method belongs to the thread class.	Wait() method belongs to the object class.
Sleep does not release the lock that the current thread holds.	wait() releases the lock which allows other threads to acquire it.

Sleep()	Wait()
This method is a static method.	This method is not a static method.
Sleep() does not throw an InterruptedException.	InterruptedException is thrown if the thread is interrupted while waiting.
Mainly used to delay a thread for some specific time duration.	Mainly used to pause a thread until notified by another thread.
<p>Sleep() Has Two Overloaded Methods:</p> <ul style="list-style-type: none"> <li>• sleep(long millis): milliseconds</li> <li>• sleep(long millis, int nanos): Nanoseconds</li> </ul>	<p>Wait() Has Three Overloaded Methods:</p> <ul style="list-style-type: none"> <li>• wait()</li> <li>• wait(long timeout)</li> <li>• wait(long timeout, int nanos)</li> </ul>

#### 47. What are the differences between String and StringBuffer?

String	StringBuffer
Store of a sequence of characters.	Provides functionality to work with the strings.
It is immutable.	It is mutable (can be modified and other string operations could be performed on them.)
No thread operations in a string.	It is thread-safe (two threads can't call the methods of StringBuffer simultaneously)

## 48. What are the differences between StringBuffer and StringBuilder?

StringBuffer	StringBuilder
StringBuffer provides functionality to work with the strings.	StringBuilder is a class used to build a mutable string.
It is thread-safe (two threads can't call the methods of StringBuffer simultaneously)	It is not thread-safe (two threads can call the methods concurrently)
Comparatively slow as it is synchronized.	Being non-synchronized, implementation is faster

## 49. Which among String or String Buffer should be preferred when there are a lot of updates required to be done in the data?

The string is preferred over StringBuffer as StringBuilder is faster than StringBuffer, but StringBuffer objects are the preferred over as it provides more thread safety.

## 50. Why is StringBuffer called mutable?

StringBuffer class in Java is used to represent a changeable string of characters. It offers an alternative to the immutable String class by enabling you to change a string's contents without constantly creating new objects. Mutable (modifiable) strings are created with the help of the StringBuffer class. The StringBuffer class in Java is identical to the String class except that it is changeable.

**Example:**

Java

```
// Java Program to demonstrate use of StringBuffer
public class StringBufferExample {
    public static void main(String[] args)
    {
        StringBuffer s = new StringBuffer();
        s.append("Geeks");
        s.append("for");
        s.append("Geeks");
        String message = s.toString();
        System.out.println(message);
    }
}
```

## Output

GeeksforGeeks

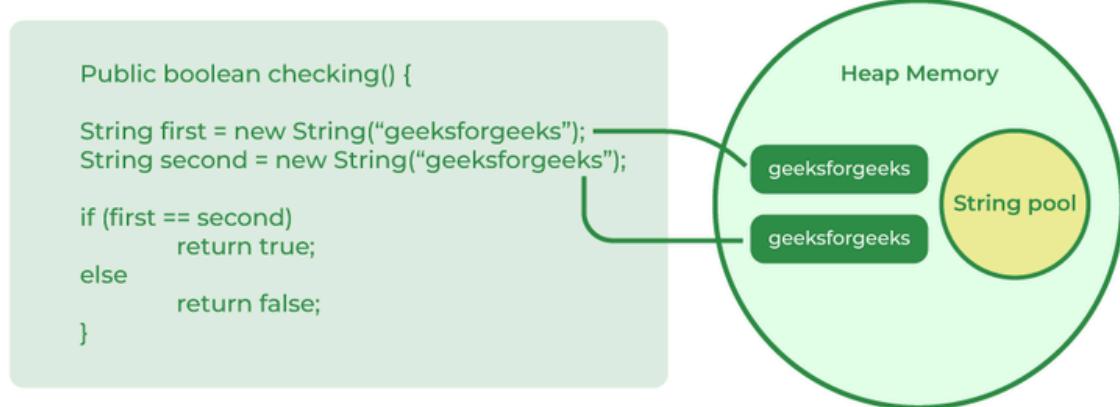
### 51. How is the creation of a String using new() different from that of a literal?

String using new() is different from the literal as when we declare string it stores the elements inside the stack memory whereas when it is declared using new() it allocates a dynamic memory in the heap memory. The object gets created in the heap memory even if the same content object is present.

#### Syntax:

```
String x = new String("ABC");
```

## String pool by means of new operator



## 52. What is an array in Java?

An Array in Java is a data structure that is used to store a fixed-size sequence of elements of the same type. Elements of an array can be accessed by their index, which starts from 0 and goes up to a length of minus 1. Array declaration in Java is done with the help of square brackets and size is also specified during the declaration.

### Syntax:

```
int[] Arr = new int[5];
```

## 53. On which memory arrays are created in Java?

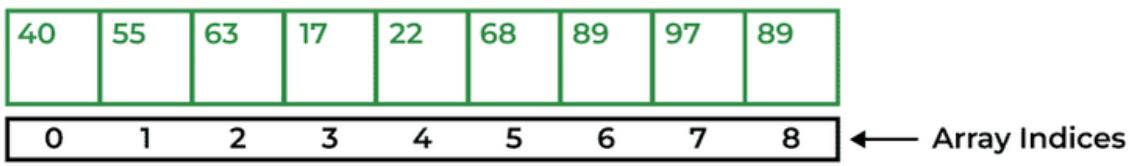
Arrays in Java are created in heap memory. When an array is created with the help of a new keyword, memory is allocated in the heap to store the elements of the array. In Java, the heap memory is managed by the Java Virtual Machine(JVM) and it is also shared between all threads of the Java Program. The memory which is no longer in use by the program, JVM uses a garbage collector to reclaim the memory.

Arrays in Java are created dynamically which means the size of the array is determined during the runtime of the program. The size of the array is specified during the declaration of the array and it cannot be changed once the array is created.

#### 54. What are the types of an array?

There are two types of arrays i.e., Primitive arrays and References Arrays.

- **Single-Dimensional Arrays:** Arrays that have only one dimension i.e., an array of integers or an array of strings are known as single-dimensional arrays.



Array Length = 9

First Index = 0

Last Index = 8

Syntax:

```
data_type[] Array_Name = new data_type[ArraySize];
```

- **Multi-Dimensional Arrays:** Arrays that have two or more dimensions such as two-dimensional or three-dimensional arrays.

#### 55. Why does the Java array index start with 0?

The index of an array signifies the distance from the start of the array. So, the first element has 0 distance therefore the starting index is 0.

### Syntax:

[Base Address + (index \* no\_of\_bytes)]

## 56. What is the difference between int array[] and int[] array?

Both int array[] and int[] array are used to declare an array of integers in java. The only difference between them is on their syntax no functionality difference is present between them.

int arr[] is a C-Style syntax to declare an Array.

int[] arr is a Java-Style syntax to declare an Array.

However, it is generally recommended to use Java-style syntax to declare an Array. As it is easy to read and understand also it is more consistent with other Java language constructs.

## 57. How to copy an array in Java?

In Java there are multiple ways to copy an Array based on the requirements.

- **clone() method in Java:** This method in Java is used to create a shallow copy of the given array which means that the new array will share the same memory as the original array.

```
int[] Arr = { 1, 2, 3, 5, 0};  
int[] tempArr = Arr.clone();
```

- **arraycopy() method:** To create a deep copy of the array we can use this method which creates a new array with the same values as the original array.

```
int[] Arr = {1, 2, 7, 9, 8};  
int[] tempArr = new int[Arr.length];  
System.arraycopy(Arr, 0, tempArr, 0, Arr.length);
```

- **copyOf() method:** This method is used to create a new array with a specific length and copies the contents of the original array to the new array.

```
int[] Arr = {1, 2, 4, 8};  
int[] tempArr = Arrays.copyOf(Arr, Arr.length);
```

- **copyOfRange() method:** This method is very similar to the copyOf() method in Java, but this method also allows us to specify the range of the elements to copy from the original array.

```
int[] Arr = {1, 2, 4, 8};  
int[] temArr = Arrays.copyOfRange(Arr, 0, Arr.length);
```

## 58. What do you understand by the jagged array?

A jagged Array in Java is just a two-dimensional array in which each row of the array can have a different length. Since all the rows in a 2-d Array have the same length but a jagged array allows more flexibility in the size of each row. This feature is very useful in conditions where the data has varying lengths or when memory usage needs to be optimized.

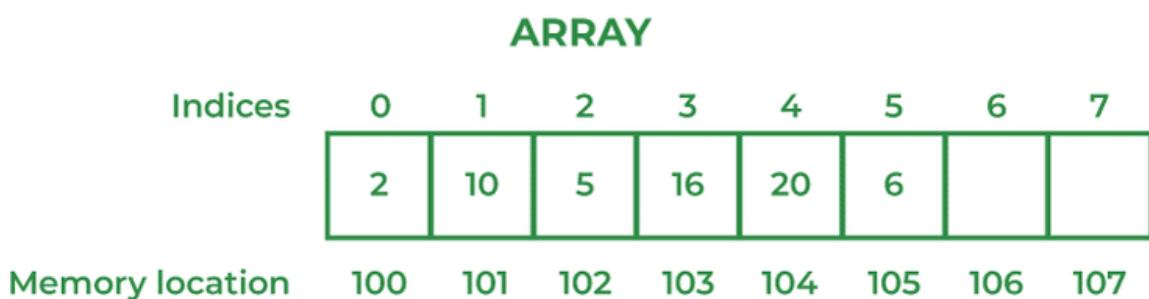
## Syntax:

```
int[][] Arr = new int[][][] {
    {1, 2, 8},
    {7, 5},
    {6, 7, 2, 6}
};
```

## 59. Is it possible to make an array volatile?

In Java, it is not possible to make an array volatile. Volatile keywords in Java can only be applied to individual variables but not to arrays or collections. The value of the Variable is always read from and written to the main memory when it is defined as volatile rather than being cached in a thread's local memory. This makes it easier to make sure that all threads that access the variable can see changes made to it.

## 60. What are the advantages and disadvantages of an array?



The advantages of Arrays are:

- Direct and effective access to any element in the collection is made possible by arrays. An array's elements can be accessed using an

$O(1)$  operation, which means that the amount of time needed to do so is constant and independent of the array's size.

- Data can be stored effectively in memory using arrays. The size of an array is known at compile time since its elements are stored in contiguous memory regions.
- Due to the fact that the data is stored in contiguous memory areas, arrays provide quick data retrieval.
- Arrays are easy to implement and understand, making them an ideal choice for beginners learning computer programming.

### Disadvantages of Arrays are:

- Arrays are created with a predetermined size that is chosen at that moment. This means that if the array's size needs to be extended, a new array will need to be made, and the data will need to be copied from the old array to the new array, which can take a lot of time and memory.
- There may be unused memory space in an array's memory space if the array is not completely occupied. If you have poor recall, this can be a problem.
- Compared to other data structures like linked lists and trees, arrays might be rigid due to their fixed size and limited support for sophisticated data types.
- Because an array's elements must all be of the same data type, it does not support complex data types like objects and structures.

## 61. What is an object-oriented paradigm?

Paradigm literally means a pattern or a method. Programming paradigms are the methods to solve a program that is of four types namely, Imperative, logical, functional, and object-oriented. When objects are used as base entities upon which the methods are applied, encapsulation or inheritance functionalities are performed, it is known as an object-oriented paradigm.

## 62. What are the main concepts of OOPs in Java?

The main concepts of OOPs in Java are mentioned below:

- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

**63. What is the difference between an object-oriented programming language and an object-based programming language?**

Object-Oriented Programming Language	Object-Based Programming Language
Object-oriented programming language covers larger concepts like inheritance, polymorphism, abstraction, etc.	The scope of object-based programming is limited to the usage of objects and encapsulation.
It supports all the built-in objects	It doesn't support all the built-in objects
Examples: Java, C#, etc.	Examples: Java script, visual basics, etc.

**64. How is the 'new' operator different from the 'newInstance()' operator in Java?**

the new operator is used to create objects, but if we want to decide the type of object to be created at runtime, there is no way we can use the new operator. In this case, we have to use the [newInstance\(\)](#). [method](#).

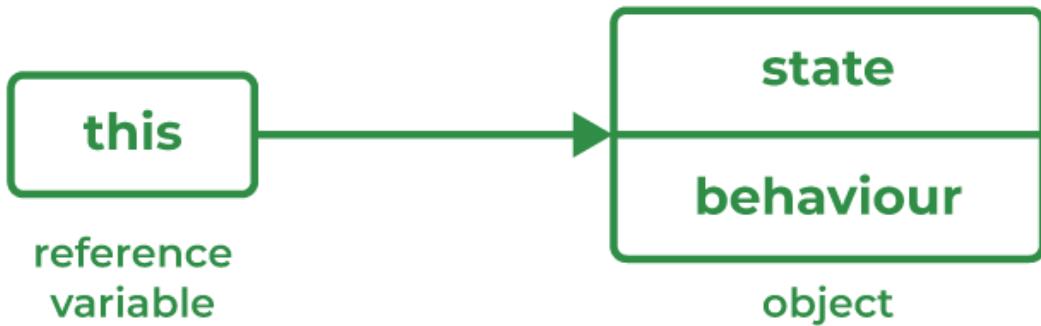
**65. What are Classes in Java?**

In Java, Classes are the collection of objects sharing similar characteristics and attributes. Classes represent the blueprint or template from which objects are created. Classes are not real-world entities but help us to create objects which are real-world entities.

## 66. What is the difference between static (class) method and instance method?

Static(Class) method	Instance method
Static method is associated with a class rather than an object.	The instance method is associated with an object rather than a class.
Static methods can be called using the class name only without creating an instance of a class.	The instance method can be called on a specific instance of a class using the object reference.
Static methods do not have access to <b>this</b> keyword.	Instance methods have access to <b>this</b> keyword.
This method can access only static members of the class	This method can access both static and non-static methods of the class.

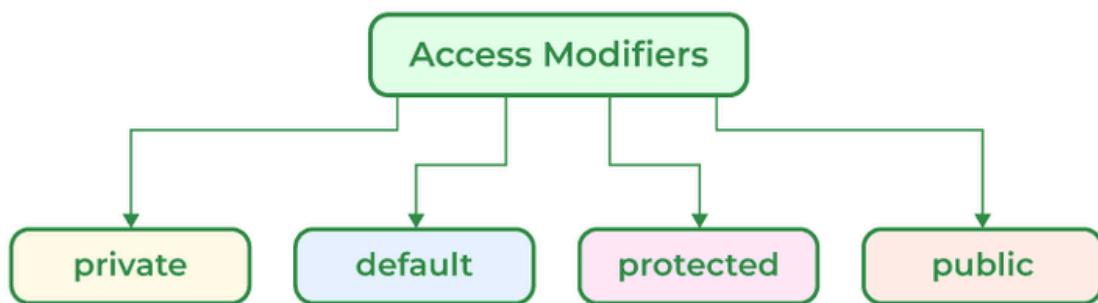
## 67. What is this keyword in Java?



'this' is a keyword used to reference a variable that refers to the current object.

## 68. What are Brief Access Specifiers and Types of Access Specifiers?

### Access Modifiers in Java



Access Specifiers in Java help to restrict the scope of a class, constructor, variable, method, or data member. There are four types of Access Specifiers in Java mentioned below:

1. Public
2. Private
3. Protected
4. Default

## 69. What will be the initial value of an object reference which is defined as an instance variable?

The initial value of an object reference which is defined as an instance variable is a NULL value.

## 70. What is an object?

The object is a real-life entity that has certain properties and methods associated with it. The object is also defined as the instance of a class. An object can be declared using a new keyword.

## 71. What are the different ways to create objects in Java?

Methods to create objects in Java are mentioned below:

1. Using new keyword
2. Using new instance
3. Using clone() method
4. Using deserialization
5. Using the newInstance() method of the Constructor class

To know more about methods to create objects in Java refer to [this article](#).

## 72. What are the advantages and disadvantages of object cloning?

There are many advantages and disadvantages of using object cloning as mentioned below:

### Advantages:

- In Java, the '=' assignment operator cannot be used for cloning as it simply creates a copy of reference variables. To overcome such discrepancy the clone() method of Object class can be used over the assignment operator.
- The clone() method is a protected method of class Object which means that only the Employee class can clone Employee objects.

This means no class other than Employee can clone Employee objects since it does not know the Employee class' attributes.

- Code size decreases as repetition decreases.
- Allows replication (kind of like prototype pattern) manually initializing each field creates large code if object is complex, its faster with cloning.

### Disadvantages:

- As the Object.clone() method is protected, so need to provide our own clone() and indirectly call Object.clone() from it.
- If we don't have any methods then we need to provide a Cloneable interface as we need to provide JVM information so that we can perform a clone() on our object. If not, we can't clone. Clone does shallow copy of fields if we just return super.clone() from clone method that can be problematic.

### 73. What are the advantages of passing this into a method instead of the current class object itself?

There are a few advantages of passing this into a method instead of the current class object itself these are:

- This is the final variable because of which this cannot be assigned to any new value whereas the current class object might not be final and can be changed.
- This can be used in the synchronized block.

### 74. What is the constructor?

Constructor is a special method that is used to initialize objects.

Constructor is called when an object is created. The name of constructor is same as of the class.

#### Example:

```
// Class Created  
class XYZ{
```

```
private int val;

// Constructor
XYZ(){
    val=0;
}
};
```

## 75. What happens if you don't provide a constructor in a class?

If you don't provide a constructor in a class in Java, the compiler automatically generates a default constructor with no arguments and no operation which is a default constructor.

## 76. How many types of constructors are used in Java?

There are two types of constructors in Java as mentioned below:

1. Default Constructor
2. Parameterized Constructor

**Default Constructor:** It is the type that does not accept any parameter value. It is used to set initial values for object attributes.

```
class_Name();
// Default constructor called
```

**Parameterized Constructor:** It is the type of constructor that accepts parameters as arguments. These are used to assign values to instance variables during the initialization of objects.

```
class_Name(parameter1, parameter2.....);
// All the values passed as parameter will be
// allocated accordingly
```

## 77. What is the purpose of a default constructor?

Constructors help to create instances of a class or can be said to create objects of a class. Constructor is called during the initialization of objects. A default constructor is a type of constructor which do not accept any parameter, So whatever value is assigned to properties of the objects are considered default values.

## 78. What do you understand by copy constructor in Java?

The copy constructor is the type of constructor in which we pass another object as a parameter because which properties of both objects seem the same, that is why it seems as if constructors create a copy of an object.

## 79. Where and how can you use a private constructor?

A private constructor is used if you don't want any other class to instantiate the object to avoid subclassing. The use private constructor can be seen as implemented in the example.

**Example:**

Java

```
// Java program to demonstrate implementation of Singleton
// pattern using private constructors.
import java.io.*;
class GFG {
    static GFG instance = null;
    public int x = 10;
    // private constructor can't be accessed outside the
    // class
    private GFG() {}
    // Factory method to provide the users with instances
    static public GFG getInstance()
    {
        if (instance == null)
            instance = new GFG();
    }
}
```

```
        return instance;
    }
}
// Driver Class
class Main {
    public static void main(String args[])
    {
        GFG a = GFG.getInstance();
        GFG b = GFG.getInstance();
        a.x = a.x + 10;
        System.out.println("Value of a.x = " + a.x);
        System.out.println("Value of b.x = " + b.x);
    }
}
```

## Output

```
Value of a.x = 20
Value of b.x = 20
```

## 80. What are the differences between the constructors and methods?

Java constructors are used for initializing objects. During creation, constructors are called to set attributes for objects apart from this few basic differences between them are:

1. Constructors are only called when the object is created but other methods can be called multiple times during the life of an object.
2. Constructors do not return anything, whereas other methods can return anything.

3. Constructors are used to setting up the initial state but methods are used to perform specific actions.

## 81. What is an Interface?

An interface in Java is a collection of static final variables and abstract methods that define the contract or agreement for a set of linked classes. Any class that implements an interface is required to implement a specific set of methods. It specifies the behavior that a class must exhibit but not the specifics of how it should be implemented.

Syntax:

```
interface
{
    // constant fields
    // methods that are abstract by default
}
```

Example:

Java

```
// Java Program to demonstrate Interface
import java.io.*;
interface Shape {
    double getArea();
    double getPerimeter();
}
class Circle implements Shape {
    private double radius;
    public Circle(double radius) { this.radius = radius; }
    public double getArea()
    {
        return Math.PI * radius * radius;
    }
    public double getPerimeter()
    {
        return 2 * Math.PI * radius;
    }
}
class GFG {
```

```
public static void main(String[] args)
{
    Circle circle = new Circle(5.0);
    System.out.println("Area of circle is "
                       + circle.getArea());
    System.out.println("Perimeter of circle is"
                       + circle.getPerimeter());
}
```

## Output

```
Area of circle is 78.53981633974483
Perimeter of circle is 31.41592653589793
```

## 82. Give some features of the Interface.

An Interface in Java programming language is defined as an abstract type used to specify the behavior of a class. An interface in Java is a blueprint of a behavior. A Java interface contains static constants and abstract methods.

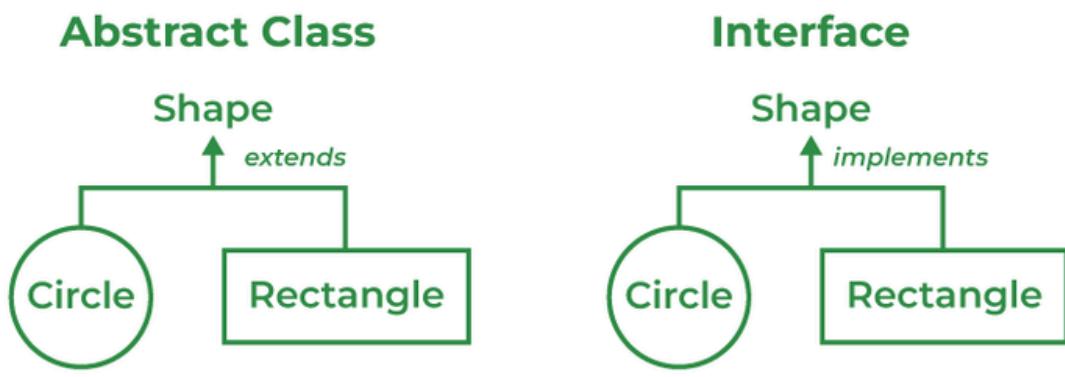
Features of the Interface are mentioned below:

- The interface can help to achieve total abstraction.
- Allows us to use multiple inheritances in Java.
- Any class can implement multiple interfaces even when one class can extend only one class.
- It is also used to achieve loose coupling.

## 83. What is a marker interface?

An Interface is recognized as an empty interface (no field or methods) it is called a marker interface. Examples of marker interfaces are Serializable, Cloneable, and Remote interfaces.

#### 84. What are the differences between abstract class and interface?

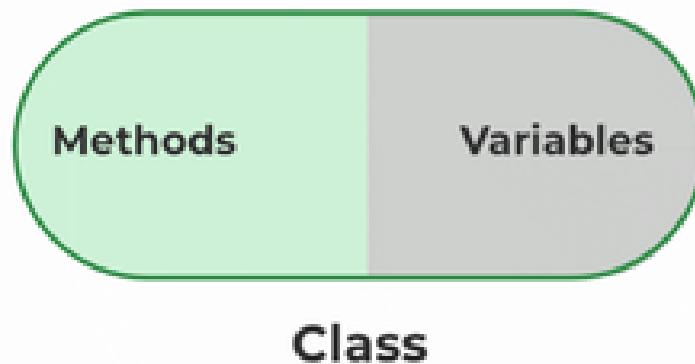


Abstract Class	Interface Class
Both abstract and non-abstract methods may be found in an abstract class.	The interface contains only abstract methods.
Abstract Class supports Final methods.	The interface class does not support Final methods.
Multiple inheritance is not supported by the Abstract class.	Multiple inheritances is supported by Interface Class.
Abstract Keyword is used to declare Abstract class.	Interface Keyword is used to declare the interface class.
<b>extend</b> keyword is used to extend an Abstract Class.	<b>implements</b> Keyword is used to implement the interface.

Abstract Class	Interface Class
Abstract Class has members like protected, private, etc.	All class members are public by default.

## 85. What do you mean by data encapsulation?

### Encapsulation in JAVA



Data Encapsulation is the concept of OOPS properties and characteristics of the classes that The interface is binded together. Basically, it bundles data and methods that operate on that data within a single unit. Encapsulation is achieved by declaring the instance variables of a class as private, which means they can only be accessed within the class.

## 86. What are the advantages of Encapsulation in Java?

The advantages of Encapsulation in Java are mentioned below:

1. Data Hiding: it is a way of restricting the access of our data members by hiding the implementation details. Encapsulation also provides a way for data hiding. The user will have no idea about the inner implementation of the class.
2. Increased Flexibility: We can make the variables of the class read-only or write-only depending on our requirements.

3. Reusability: Encapsulation also improves the re-usability and is easy to change with new requirements.
4. Testing code is easy: Code is made easy to test for unit testing.

## 87. What is the primary benefit of Encapsulation?

The main advantage of Encapsulation in Java is its ability to protect the internal state of an object from external modification or access. It is the way of hiding the implementation details of a class from outside access and only exposing a public interface that can be used to interact with the class. The main benefit is of providing a way to control and manage the state and the behavior of an object and also protecting it from modification and unauthorized access at the same time.

**Example:**

**Java**

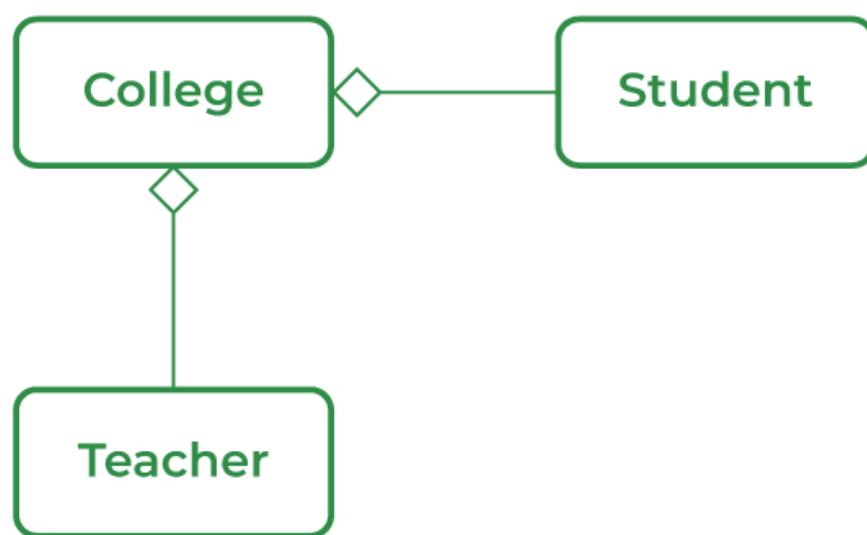
```
// Java Program to demonstrate use of Encapsulation
import java.io.*;
class Person {
    private String Name;
    private int age;
    public String getName() { return Name; }
    public void setName(String Name) { this.Name = Name; }
    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }
}
// Driver class
class GFG {
    // main function
    public static void main(String[] args)
    {
        Person p = new Person();
        p.setName("Rohan");
        p.setAge(29);
        System.out.println("Name is " + p.getName());
        System.out.println("Age is " + p.getAge());
    }
}
```

**Output**

Name is Rohan

Age is 29

### 88. What do you mean by aggregation?



Aggregation is a term related to the relationship between two classes best described as a “has-a” relationship. This kind is the most specialized version of association. It is a unidirectional association means it is a one-way relationship. It contains the reference to another class and is said to have ownership of that class.

### 89. What is the ‘IS-A’ relationship in OOPs Java?

‘IS-A’ is a type of relationship in OOPs Java where one class inherits another class.

### 90. Define Inheritance.

When an object that belongs to a subclass acquires all the properties and behavior of a parent object that is from the superclass, it is known as inheritance. A class within a class is called the subclass and the latter is referred to as the superclass. Sub class or the child class is said to be specific whereas the superclass or the parent class is generic. Inheritance provides code reusability.

## 91. What are the different types of inheritance in Java?

Inheritance is the method by which the Child class can inherit the features of the Super or Parent class. In Java, Inheritance is of four types:

- **Single Inheritance:** When a child or subclass extends only one superclass, it is known to be single inheritance. Single-parent class properties are passed down to the child class.
- **Multilevel Inheritance:** When a child or subclass extends any other subclass a hierarchy of inheritance is created which is known as multilevel inheritance. In other words, one subclass becomes the parent class of another.
- **Hierarchical Inheritance:** When multiple subclasses derive from the same parent class is known as Hierarchical Inheritance. In other words, a class that has a single parent has many subclasses.
- **Multiple Inheritance:** When a child class inherits from multiple parent classes is known as Multiple Inheritance. In Java, it only supports multiple inheritance of interfaces, not classes.

## 92. What is multiple inheritance? Is it supported by Java?

A component of the object-oriented notion known as multiple inheritances allows a class to inherit properties from many parent classes. When methods with the same signature are present in both superclasses and subclasses, an issue arises. The method's caller cannot specify to the compiler which class method should be called or even which class method should be given precedence.

*Note: Java doesn't support Multiple Inheritance*

Example:

### Java

```
// Java Program to show multiple Inheritance
import java.io.*;
interface Animal {
    void eat();
}
interface Mammal {
    void drink();
}
class Dog implements Animal, Mammal {
    public void eat() { System.out.println("Eating"); }
    public void drink() { System.out.println("Drinking"); }
    void bark() { System.out.println("Barking"); }
}
class GFG {
    public static void main(String[] args)
    {
        Dog d = new Dog();
        d.eat();
        d.drink();
        d.bark();
    }
}
```

Output

Eating  
Drinking  
Barking

### 93. How is inheritance in C++ different from Java?

Inheritance in C++	Inheritance in Java
C++ lets the user to inherit multiple classes.	Java doesn't support multiple inheritances.
When a class is created in C++, it doesn't inherit from the object class, instead exists on its own.	Java is always said to have a single inheritance as all the classes inherit in one or the other way from the object class.

### 94. Is there any limitation to using Inheritance?

Yes, there is a limitation of using Inheritance in Java, as because of inheritance one can inherit everything from super class and interface because of which subclass is too clustered and sometimes error-prone when dynamic overriding or dynamic overloading is done in certain situations.

### 95. Although inheritance is a popular OOPs concept, it is less advantageous than composition. Explain.

Inheritance is a popular concept of Object-Oriented Programming (OOP), in which a class can inherit the properties and methods from any other class, which is referred to as a Parent or superclass. On the other hand in Composition, a class can contain an instance of another class as a member variable which is often referred to as part or a component. Below are some reasons why composition is more advantageous than inheritance:

- **Tight Coupling:** Whenever any changes are made to the superclass, these changes can affect the behavior of all its child or

Subclasses. This problem makes code less flexible and also creates issues during maintenance. This problem also leads to the Tight coupling between the classes.

- **Fragile Base Class Problem:** When the changes to the base class can break the functionality of its derived classes. This problem can make it difficult to add new features or modify the existing ones. This problem is known as the Fragile Base class problem.
- **Limited Reuse:** Inheritance in Java can lead to limited code reuse and also code duplication. As a subclass inherits all the properties and methods of its superclass, sometimes it may end up with unnecessary code which is not needed. This leads to a less maintainable codebase.

## 96. What is an association?

The association is a relation between two separate classes established through their Objects. It represents Has-A's relationship.

## 97. What do you mean by aggregation?

Composition is a restricted form of Aggregation in which two entities are highly dependent on each other. It represents **part-of** the relationship.

## 98. What is the composition of Java?

Composition implies a relationship where the child **cannot exist independently** of the parent. For example Human heart, the heart doesn't exist separately from a Human.

## 99. State the difference between Composition and Aggregation.

Aggregation	Composition
It defines a “has a” relationship	It represents the part-of

Aggregation	Composition
between the objects	relationship
Objects are independent of each other.	Objects are dependent on each other.
Represent it by using the filled diamond.	Represent it by using the empty diamond.
Child objects don't have a lifetime.	Child objects have a lifetime.

## 100. Can the constructor be inherited?

No, we can't inherit a constructor.

## 101. What is Polymorphism?

Polymorphism is defined as the ability to take more than one form. It is of two types namely, Compile time polymorphism or method overloading- a function called during compile time. For instance, take a class 'area'. Based on the number of parameters it may calculate the area of a square, triangle, or circle. Run time polymorphism or method overriding- links during run time. The method inside a class overrides the method of the parent class.

## 102. What is runtime polymorphism or dynamic method dispatch?

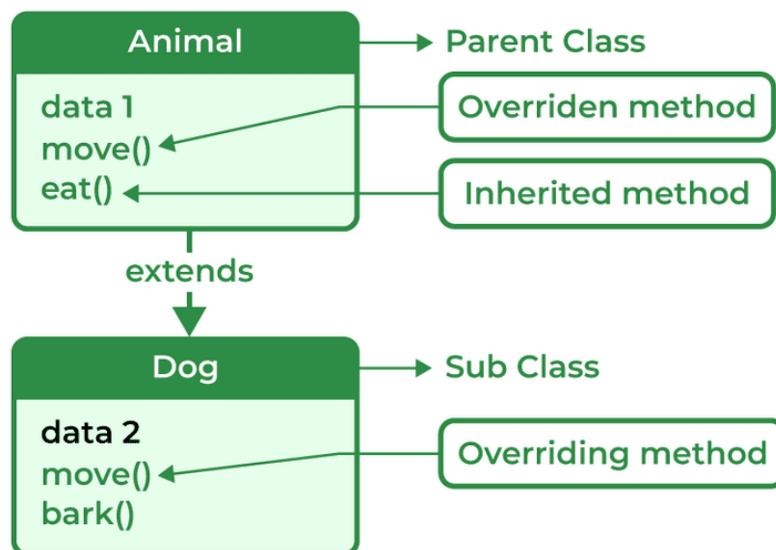
Dynamic method dispatch is a resolving mechanism for method overriding during the run time. Method overriding is the one where the method in a subclass has the same name, parameters, and return type as a method in the superclass. When the over-ridden method is called through a superclass reference, java determines which version (superclass or subclass) of that method is to be executed based upon

the type of an object being referred to at the time the call occurs. Thus the decision is made at run time. This is referred to as dynamic method dispatch.

### 103. What is method overriding?

Method overriding, also known as run time polymorphism is one where the child class contains the same method as the parent class. For instance, we have a method named 'gfg()' in the parent class. A method gfg() is again defined in the sub-class. Thus when gfg() is called in the subclass, the method within the class is executed. Here, gfg() within the class overridden the method outside.

### 104. What is method overloading?



Method overriding is a method to achieve Run-time polymorphism in Java. Method overriding is a feature that allows a child class to provide a specific implementation of a method that is already provided by one of its parent classes. When a method in a child class has the same name, the same parameters or signature, and the same return type(or sub-type) as a method in its parent class, then the method in the subclass is said to override the method in the superclass.

### 105. Can we override the static method?

No, as static methods are part of the class rather than the object so we can't override them.

### 106. Can we override the overloaded method?

Yes, since the overloaded method is a completely different method in the eyes of the compiler. Overriding isn't the same thing at all. The decision as to which method to call is deferred to runtime.

### 107. Can we overload the main() method?

Yes in Java we can overload the main method to call the main method with the help of its predefined calling method.

### 108. What are method overloading and method overriding?

**Method Overloading:** It is also known as Compile Time Polymorphism. In method overloading two or more methods are shared in the same class with a different signature.

Example:

Java

```
// Java Program to demonstrate use of Method Overloading
import java.io.*;
class GFG {
    static int multiply(int a, int b) { return a * b; }
    static int multiply(int a, int b, int c)
    {
        return a * b * c;
    }
    static int multiply(int a, int b, int c, int d)
    {
        return a * b * c * d;
    }
    public static void main(String[] args)
    {
        System.out.println("multiply() with 2 parameters");
        System.out.println(multiply(4, 5));
        System.out.println("multiply() with 3 parameters");
        System.out.println(multiply(2, 3, 4));
        System.out.println("multiply() with 4 parameters");
        System.out.println(multiply(2, 3, 4, 1));
    }
}
```

```
}
```

## Output

```
multiply() with 2 parameters
20
multiply() with 3 parameters
24
multiply() with 4 parameters
24
```

**Method Overriding:** Method Overriding occurs when a subclass can provide the implementation of a method which is already defined in the parent class or superclass. The return type, name and arguments must be similar to the methods in superclass.

### Example:

#### Java

```
// Java Program to demonstrate use of Method Overriding
import java.io.*;
class Vehicle {
    void drive()
    {
        System.out.println("drive() method of base class");
        System.out.println("driving the Car.");
    }
}
class Car extends Vehicle {
    void drive()
    {
```

```

        System.out.println(
            "drive() method of derived class");
        System.out.println("Car is driving.");
    }
}

class GFG {
    public static void main(String[] args)
    {
        Car c1 = new Car();
        Vehicle v1 = new Vehicle();
        c1.drive();
        v1.drive();
        Vehicle vehicle = new Car();
        // drive() method of Vehicle class is overridden by
        // Car class drive()
        vehicle.drive();
    }
}

```

## Output

```

drive() method of derived class
Car is driving.

drive() method of base class
driving the Car.

drive() method of derived class
Car is driving.

```

Method Overloading	Method Overriding
When two or multiple methods are in the same class with	When a subclass provides its own implementation of a method that is already defined in the parent class.

Method Overloading	Method Overriding
different parameters but the same name.	
Method overloading can only happen in the same class or between a subclass or parent class.	Method overriding can only happen in Subclass.
When an error occurs it is caught at the compile time of the program.	When an error occurs it is caught at Runtime of the program.
Example of Compile Time Polymorphism.	Example of Run Time Polymorphism.
Method Overloading may or may not require Inheritance.	Method overriding always needs Inheritance.
It occurs within the class.	It is performed in two classes with an inheritance relationship.

### 109. Can we override the private methods?

It is not possible to override the private methods in Java. Method overriding is where the method in the subclass is implemented instead of the method from the parent class. The private methods are accessible only within the class in which it is declared. Since this method is not visible to other classes and cannot be accessed, it cannot be overridden.

### 110. Can we change the scope of the overridden method in the subclass?

In Java, it is not possible to modify the overridden method's scope. The subclass method's scope must be equal to or wider than the Superclass method's overridden method's scope. The overridden method in the subclass, for instance, can have a public scope or a more accessible scope like protected or default if the overridden method in the superclass has a public scope. It cannot, however, have a more exclusive scope like private.

### **111. Can we modify the throws clause of the superclass method while overriding it in the subclass?**

We can modify the throws clause of the Superclass method with some limitations, we can change the throws clause of the superclass method while overriding it in the subclass. The subclass overridden method can only specify unchecked exceptions if the superclass method does not declare any exceptions. If the superclass method declares an exception, the subclass method can declare the same exception, a subclass exception, or no exception at all. However, the subclass method cannot declare a parent exception that is broader than the ones declared in the superclass method.

### **112. Can you have virtual functions in Java?**

Yes, Java supports virtual functions. Functions are by default virtual and can be made non-virtual using the final keyword.

### **113. What is Abstraction?**

Abstraction refers to the act of representing essential features without including background details. The detailed information or the implementation is hidden. The most common example of abstraction is a car, we know how to turn on the engine, accelerate and move, however, the way engine works, and its internal components are complex logic hidden from the general users. This is usually done to handle the complexity.

## 114. What is Abstract class?

A class declared as abstract, cannot be instantiated i.e., the object cannot be created. It may or may not contain abstract methods but if a class has at least one abstract method, it must be declared abstract.

*Example of an abstract class with abstract method:*

Java

```
// Java Program to implement
// abstract method
import java.io.*;
// Abstract class
abstract class Fruits {
    abstract void run();
}
// Driver Class
class Apple extends Fruits {
    void run()
    {
        System.out.println("Abstract class example");
    }
    // main method
    public static void main(String args[])
    {
        Fruits obj = new Apple();
        obj.run();
    }
}
```

## 115. When Abstract methods are used?

An abstract method is used when we want to use a method but want to child classes to decide the implementation in that case we use Abstract methods with the parent classes.

## 116. How can you avoid serialization in the child class if the base class is implementing the Serializable interface?

Serialization in the child class if the base class is implementing the Serializable interface then we can avoid it by defining the

writeObject() method and throwing NotSerializableException().

## 117. What is Collection Framework in Java?

Collections are units of objects in Java. The collection framework is a set of interfaces and classes in Java that are used to represent and manipulate collections of objects in a variety of ways. The collection framework contains classes(ArrayList, Vector, LinkedList, PriorityQueue, TreeSet) and multiple interfaces (Set, List, Queue, Deque) where every interface is used to store a specific type of data.

## 118. Explain various interfaces used in the Collection framework.

Collection framework implements

1. Collection Interface
2. List Interface
3. Set Interface
4. Queue Interface
5. Deque Interface
6. Map Interface

**Collection interface:** Collection is the primary interface available that can be imported using java.util.Collection.

**Syntax:**

```
public interface Collection<E> extends iterable
```

## 119. How can you synchronize an ArrayList in Java?

An ArrayList can be synchronized using two methods mentioned below:

1. Using Collections.synchronizedList()
2. Using CopyOnWriteArrayList

### Using Collections.synchronizedList():

```
public static List<T> synchronizedList(List<T> list)
```

### Using CopyOnWriteArrayList:

1. Create an empty List.
2. It implements the List interface
3. It is a thread-safe variant of ArrayList
4. T represents generic

## 120. Why do we need a synchronized ArrayList when we have Vectors (which are synchronized) in Java?

ArrayList is in need even when we have Vectors because of certain reasons:

1. ArrayList is faster than Vectors.
2. ArrayList supports multithreading whereas Vectors only supports single-thread use.
3. ArrayList is safer to use, as Vectors supports single threads and individual operations are less safe and take longer to synchronize.
4. Vectors are considered outdated in Java because of their synchronized nature.

## 121. Why can't we create a generic array?

Generic arrays can't be created because an **array** carries type information of its elements at runtime because of which during runtime it throw 'ArrayStoreException' if the elements' type is not similar. Since generics type information gets erased at compile time by Type Erasure, the array store check would have been passed where it should have failed.

## 122. Contiguous memory locations are usually used for storing actual values in an array but not in ArrayList. Explain.

The elements of an array are stored in contiguous memory locations, which means that each element is stored in a separate block based on its location within the array. Since the elements of the array are stored in contiguous locations, it can be relatively easy to access any element by its index, as the element address can be calculated based on the location of the element. But Java implements ArrayLists as dynamic arrays, which means that the size can change as elements are removed or added. ArrayList elements are not stored in contiguous memory locations in order to accommodate this dynamic nature. Instead, the ArrayList makes use of a method known as an expandable array in which the underlying array is expanded to a larger size as needed and the elements are then copied to the new location. In contrast to an ArrayList, which has a dynamic size and does not store its elements in contiguous memory locations, an array has a fixed size and its elements are stored there.

### 123. Explain the method to convert ArrayList to Array and Array to ArrayList.

#### Conversion of List to ArrayList

There are multiple methods to convert List into ArrayList



#### Methods:

1. **Arrays.asList()**
2. **Collections.addAll()**
3. **add() method**

Programmers can convert an Array to ArrayList using asList() method of the Arrays class. It is a static method of the Arrays class that accepts the List object.

## Syntax:

```
Arrays.asList(item)
```

## Example:

Java

```
// Java program to demonstrate conversion of
// Array to ArrayList of fixed-size.
import java.util.*;
// Driver Class
class GFG {
    // Main Function
    public static void main(String[] args)
    {
        String[] temp = { "Abc", "Def", "Ghi", "Jkl" };
        // Conversion of array to ArrayList
        // using Arrays.asList
        List conv = Arrays.asList(temp);
        System.out.println(conv);
    }
}
```

## Output

```
[Abc, Def, Ghi, Jkl]
```

## Conversion of ArrayList to Array



### Methods:

1. `Object[]toArray()`
2. `T[]toArray(T[]a)`
3. `get()` method

Java programmers can convert `ArrayList` to `Array`.

### Syntax:

```
List_object.toArray(new String[List_object.size()])
```

### Example:

#### Java

```
// Java program to demonstrate working of
// Object[] toArray()
import java.io.*;
import java.util.List;
import java.util.ArrayList;
// Driver Class
class GFG {
    // Main Function
    public static void main(String[] args)
    {
        // List declared
        List<Integer> arr = new ArrayList<Integer>();
        arr.add(1);
        arr.add(2);
        arr.add(3);
        arr.add(2);
        arr.add(1);
        // Conversion
        Object[] objects = arr.toArray();
        // Printing array of objects
        for (Object obj : objects)
            System.out.print(obj + " ");
```

```
}
```

## Output

```
1 2 3 2 1
```

**124. How does the size of ArrayList grow dynamically? And also state how it is implemented internally.**

Due to ArrayLists array-based nature, it grows dynamically in size ensuring that there is always enough room for elements. When an ArrayList element is first created, the default capacity is around 10-16 elements which basically depends on the Java version. ArrayList elements are copied over from the original array to the new array when the capacity of the original array is full. As the ArrayList size increases dynamically, the class creates a new array of bigger sizes and it copies all the elements from the old array to the new array. Now, the reference of the new array is used internally. This process of dynamically growing an array is known as resizing.

**125. What is a Vector in Java?**

Vectors in Java are similar and can store multiple elements inside them. Vectors follow certain rules mentioned below:

1. Vector can be imported using `Java.util.Vector`.

2. Vector is implemented using a dynamic array as the size of the vector increases and decreases depending upon the elements inserted in it.
3. Elements of the Vector using index numbers.
4. Vectors are synchronized in nature means they only used a single thread ( only one process is performed at a particular time ).
5. The vector contains many methods that are not part of the collections framework.

### Syntax:

```
Vector gfg = new Vector(size, increment);
```

## 126. How to make Java ArrayList Read-Only?

An ArrayList can be made ready only using the method provided by Collections using the Collections.unmodifiableList() method.

### Syntax:

```
array_READONLY = Collections.unmodifiableList(ArrayList);
```

### Example:

#### Java

```

// Java program to demonstrate
// unmodifiableList() method
import java.util.*;
public class Main {
    public static void main(String[] argv) throws Exception
    {
        try {
            // creating object of ArrayList
            <Character> ArrayList<Character>
            temp
            = new ArrayList<Character>();
            // populate the list
        }
    }
}
```

```
temp.add('X');
temp.add('Y');
temp.add('Z');
// printing the list
System.out.println("Initial list: " + temp);
// getting readonly list
// using unmodifiableList() method
List<Character>
    new_array
    = Collections.unmodifiableList(temp);
// printing the list
System.out.println("ReadOnly ArrayList: "
    + new_array);
// Adding element to new Collection
System.out.println("\nIf add element in "
    + " the ReadOnly ArrayList");
new_array.add('A');
}
catch (UnsupportedOperationException e) {
    System.out.println("Exception is thrown : "
        + e);
}
}
}
```

## Output

```
Initial list: [X, Y, Z]
ReadOnly ArrayList: [X, Y, Z]
```

```
If add element in the ReadOnly ArrayList
Exception is thrown : java.lang.UnsupportedOperationException
```

## 127. What is a priority queue in Java?

## Priority Queue

Initial Queue = {}

Operation	Return value	Queue Content
insert ( C )		C
insert ( O )		C O
insert ( D )		C O D
<b>remove max</b>	O	C D
insert ( I )		C D I
insert ( N )		C D I N
<b>remove max</b>	N	C D I
insert ( G )		C D I G

A priority queue is an abstract data type similar to a regular queue or stack data structure. Elements stored in elements are depending upon the priority defined from low to high. The PriorityQueue is based on the priority heap.

Syntax:

Java

```
// Java program to demonstrate the
// working of PriorityQueue
import java.util.*;
class PriorityQueueDemo {
    // Main Method
    public static void main(String args[])
    {
        // Creating empty priority queue
        PriorityQueue<Integer>
            var1
            = new PriorityQueue<Integer>();
        // Adding items to the pQueue using add()
        var1.add(10);
        var1.add(20);
        var1.add(15);
        // Printing the top element of PriorityQueue
        System.out.println(var1.peek());
    }
}
```

Output

10

## 128. Explain the LinkedList class.

LinkedList class is Java that uses a doubly linked list to store elements. It inherits the AbstractList class and implements List and Deque interfaces. Properties of the LinkedList Class are mentioned below:

1. LinkedList classes are non-synchronized.
2. Maintains insertion order.
3. It can be used as a list, stack, or queue.

Syntax:

```
LinkedList<class> list_name=new LinkedList<class>();
```

## 129. What is the Stack class in Java and what are the various methods provided by it?

A Stack class in Java is a LIFO data structure that implements the Last In First Out data structure. It is derived from a Vector class but has functions specific to stacks. The Stack class in java provides the following methods:

- **peek():** returns the top item from the stack without removing it
- **empty():** returns true if the stack is empty and false otherwise
- **push():** pushes an item onto the top of the stack

- **pop()**: removes and returns the top item from the stack
- **search()**: returns the 1, based position of the object from the top of the stack. If the object is not in the stack, it returns -1

### 130. What is Set in the Java Collections framework and list down its various implementations?

Sets are collections that don't store duplicate elements. They don't keep any order of the elements. The Java Collections framework provides several implementations of the Set interface, including:

- **HashSet**: HashSet in Java, stores the elements in a hash table which provides faster lookups and faster insertion. HashSet is not ordered.
- **LinkedHashSet**: LinkedHashSet is an implementation of HashSet which maintains the insertion order of the elements.
- **TreeSet**: TreeSet stores the elements in a sorted order that is determined by the natural ordering of the elements or by a custom comparator provided at the time of creation.

### 131. What is the HashSet class in Java and how does it store elements?

The HashSet class implements the Set interface in the Java Collections Framework and is a member of the HashSet class. Unlike duplicate values, it stores a collection of distinct elements. In this implementation, each element is mapped to an index in an array using a hash function, and the index is used to quickly access the element. It produces an index for the element in the array where it is stored based on the input element. Assuming the hash function distributes the elements among the buckets appropriately, the HashSet class provides constant-time performance for basic operations (add, remove, contain, and size).

### 132. What is LinkedHashSet in Java Collections Framework?

The LinkedHashSet is an ordered version of HashSet maintained by a doubly-linked List across all the elements. It is very helpful when

iteration order is needed. During Iteration in LinkedHashSet, elements are returned in the same order they are inserted.

### Syntax:

```
LinkedHashSet<E> hs = new LinkedHashSet<E>();
```

### Example:

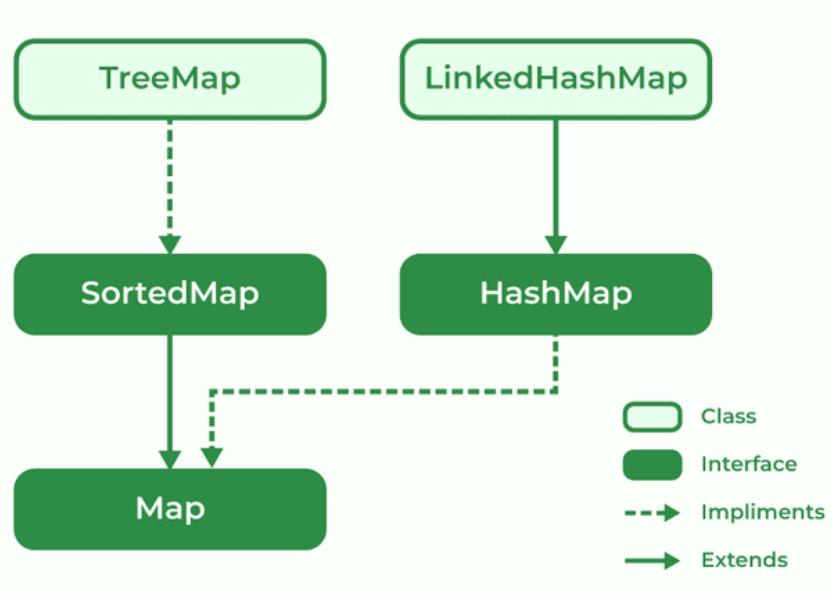
#### Java

```
// Java Program to implement  
// LinkedHashSet  
import java.io.*;  
import java.util.*;  
// Driver Class  
class GFG {  
    // Main Function  
    public static void main(String[] args)  
    {  
        // LinkedHashSet declared  
        LinkedHashSet<Integer> hs = new LinkedHashSet<Integer>();  
        // Add elements in HashSet  
        hs.add(1);  
        hs.add(2);  
        hs.add(5);  
        hs.add(3);  
        // Print values  
        System.out.println("Values: " + hs);  
    }  
}
```

### Output

Values:[1, 2, 5, 3]

### 133. What is a Map interface in Java?



The map interface is present in the Java collection and can be used with `Java.util` package. A map interface is used for mapping values in the form of a key-value form. The map contains all unique keys. Also, it provides methods associated with it like `containsKey()`, `contains value ()`, etc.

There are multiple types of maps in the map interface as mentioned below:

1. `SortedMap`
2. `TreeMap`
3. `HashMap`
4. `LinkedHashMap`

### 134. Explain Treemap in Java

`TreeMap` is a type of map that stores data in the form of key-value pair. It is implemented using the red-black tree. Features of `TreeMap` are :

1. It contains only unique elements.
2. It cannot have a NULL key

3. It can have multiple NULL values.
4. It is non-synchronized.
5. It maintains ascending order.

### 135. What is EnumSet?

EnumSet is a specialized implementation of the Set interface for use with enumeration type. A few features of EnumSet are:

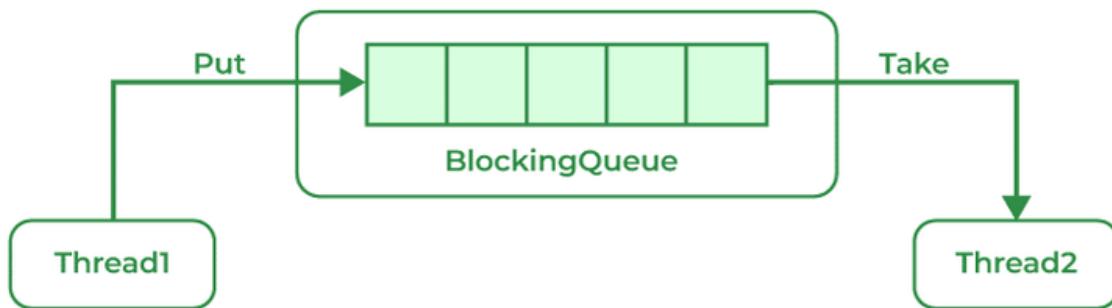
1. It is non-synchronized.
2. Faster than HashSet.
3. All of the elements in an EnumSet must come from a single enumeration type.
4. It doesn't allow null Objects and throws NullPointerException for exceptions.
5. It uses a fail-safe iterator.

#### Syntax:

```
public abstract class EnumSet<E extends Enum<E>>
```

**Parameter:** E specifies the elements.

### 136. What is BlockingQueue?



A blocking queue is a Queue that supports the operations that wait for the queue to become non-empty while retrieving and removing the element, and wait for space to become available in the queue while adding the element.

**Syntax:**

```
public interface BlockingQueue<E> extends Queue<E>
```

**Parameters:** E is the type of elements stored in the Collection

**137. What is the ConcurrentHashMap in Java and do you implement it?**

ConcurrentHashMap is implemented using Hashtable.

**Syntax:**

```
public class ConcurrentHashMap<K, V>
extends AbstractMap<K, V>
implements ConcurrentMap<K, V>, Serializable
```

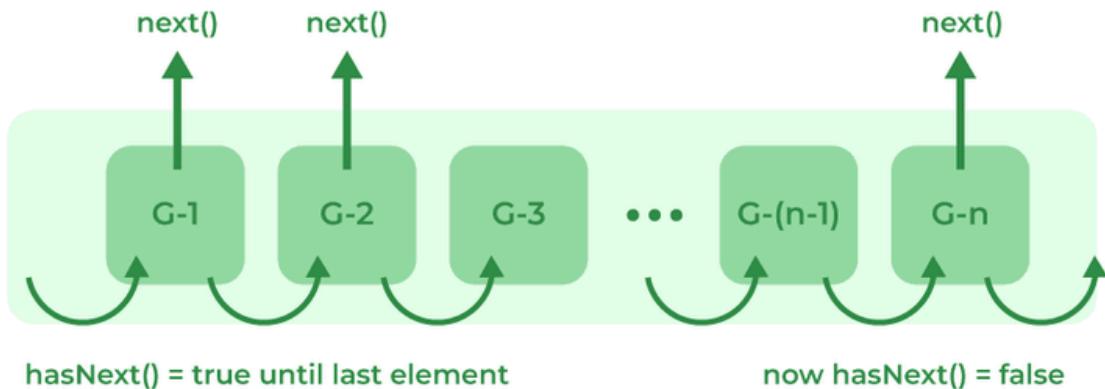
**Parameters:** K is the key Object type and V is the value Object type

**138. Can you use any class as a Map key?**

Yes, we can use any class as a Map Key if it follows certain predefined rules mentioned below:

1. The class overriding the equals() method must also override the hashCode() method
2. The concurrentHashMap class is thread-safe.
3. The default concurrency level of ConcurrentHashMap is 16.
4. Inserting null objects in ConcurrentHashMap is not possible as a key or as value.

## 139. What is an Iterator?



### Java Iterator : Forward Direction

The Iterator interface provides methods to iterate over any Collection in Java. Iterator is the replacement of Enumeration in the Java Collections Framework. It can get an iterator instance from a Collection using the `_iterator()` method. It also allows the caller to remove elements from the underlying collection during the iteration.

## 140. What is an enumeration?

Enumeration is a user-defined data type. It is mainly used to assign names to integral constants, the names make a program easy to read and maintain. The main objective of the enum is to define user-defined data types.

### Example:

```
// A simple enum example where enum is declared
// outside any class (Note enum keyword instead of
// class keyword)
enum Color
{
    RED, GREEN, BLUE;
}
```

## 141. What is the difference between Collection and Collections?

Collection	Collections
The Collection is an Interface.	Collections is a class.
It provides the standard functionality of data structure.	It is to sort and synchronize the collection elements.
It provides the methods that can be used for the data structure.	It provides static methods that can be used for various operations.

## 142. Differentiate between Array and ArrayList in Java.

Array	ArrayList
Single-dimensional or multidimensional	Single-dimensional
For and for each used for iteration	Here iterator is used to traverse riverArrayList
length keyword returns the size of the array.	size() method is used to compute the size of ArrayList.
The array has Fixed-size.	ArrayList size is dynamic and can be increased or decreased in size when required.
It is faster as above we see it of fixed size	It is relatively slower because of its dynamic nature

Array	ArrayList
Primitive data types can be stored directly in unlikely objects.	Primitive data types are not directly added to unlikely arrays, they are added indirectly with help of autoboxing and unboxing
They can not be added here hence the type is in the unsafe.	They can be added here hence makingArrayList type-safe.
The assignment operator only serves the purpose	Here a special method is used known as add() method

### 143. What is the difference between Array and Collection in Java?

Array	Collections
Array in Java has a fixed size.	Collections in Java have dynamic sizes.
In an Array, Elements are stored in contiguous memory locations.	In Collections, Elements are not necessarily stored in contiguous memory locations.
Objects and primitive data types can be stored in an array.	We can only store objects in collections.
Manual manipulation is required for resizing the array.	Resizing in collections is handled automatically.
The array has basic methods for manipulation.	Collections have advanced methods for manipulation and

Array	Collections
	iteration.
The array is available since the beginning of Java.	Collections were introduced in Java 1.2.

#### 144. Difference between ArrayList and LinkedList.

ArrayList	LinkedList
ArrayList is Implemented as an expandable Array.	LinkedList is Implemented as a doubly-linked list.
In ArrayList, Elements are stored in contiguous memory locations	LinkedList Elements are stored in non-contiguous memory locations as each element has a reference to the next and previous elements.
ArrayLists are faster for random access.	LinkedLists are faster for insertion and deletion operations
ArrayLists are more memory efficient.	LinkedList is less memory efficient
ArrayLists Use more memory due to maintaining the array size.	LinkedList Uses less memory as it only has references to elements
The search operation is faster in ArrayList.	The search operation is slower in LinkedList

#### 145. Differentiate between ArrayList and Vector in Java.

ArrayList	Vector
ArrayLists are implemented as an expandable array.	Vector is Implemented as a growable array.
ArrayList is not synchronized.	The vector is synchronized.
ArrayLists are Faster for non-concurrent operations.	Vector is Slower for non-concurrent operations due to added overhead of synchronization.
ArrayLists were Introduced in Java 1.2.	Vector was Introduced in JDK 1.0.
Recommended for use in a single-threaded environment.	Vectors are Recommended for use in a multi-threaded environment.
The default initial capacity of ArrayLists is 10.	In Vectors, the default initial capacity is 10 but the default increment is twice the size.
ArrayList performance is high.	Vector performance is low.

#### 146. What is the difference between Iterator and ListIterator?

Iterator	ListIterator
Can traverse elements present in Collection only in the forward direction.	Can traverse elements present in Collection both in forward and backward directions.

Iterator	ListIterator
Used to traverse Map, List, and Set.	Can only traverse List and not the other two.
Indexes can't be obtained using Iterator	It has methods like nextIndex() and previousIndex() to obtain indexes of elements at any time while traversing the List.
Can't modify or replace elements present in Collection	Can modify or replace elements with the help of set(E e)
Can't add elements, and also throws ConcurrentModificationException.	Can easily add elements to a collection at any time.
Certain methods of Iterator are next(), remove(), and hasNext().	Certain methods of ListIterator are next(), previous(), hasNext(), hasPrevious(), add(E e).

#### 147. Differentiate between HashMap and HashTable.

HashMap	HashTable
HashMap is not synchronized	HashTable is synchronized
One key can be a NULL value	NULL values not allowed
The iterator is used to traverse HashMap.	Both Iterator and Enumerar can be used

HashMap	HashTable
HashMap is faster.	HashTable is slower as compared to HashMap.

#### 148. What is the difference between Iterator and Enumeration?

Iterator	Enumeration
The Iterator can traverse both legacies as well as non-legacy elements.	Enumeration can traverse only legacy elements.
The Iterator is fail-fast.	Enumeration is not fail-fast.
The Iterators are slower.	Enumeration is faster.
The Iterator can perform a remove operation while traversing the collection.	The Enumeration can perform only traverse operations on the collection.

#### 149. What is the difference between Comparable and Comparator?

Comparable	Comparator
The interface is present in java.lang package.	The Interface is present in java.util package.
Provides compareTo() method to sort elements.	Provides compare() method to sort elements.

Comparable	Comparator
It provides single sorting sequences.	It provides multiple sorting sequences.
The logic of sorting must be in the same class whose object you are going to sort.	The logic of sorting should be in a separate class to write different sorting based on different attributes of objects.
Method sorts the data according to fixed sorting order.	Method sorts the data according to the customized sorting order.
It affects the original class.	It doesn't affect the original class.
Implemented frequently in the API by Calendar, Wrapper classes, Date, and String.	It is implemented to sort instances of third-party classes.

## 150. What is the difference between Set and Map?

Set	Map
The Set interface is implemented using java.util package.	The map is implemented using java.util package.
It can extend the collection interface.	It does not extend the collection interface.
It does not allow duplicate values.	It allows duplicate values.
The set can sort only one null	The map can sort multiple null

Set	Map
value.	values.

## Java Intermediate Interview Questions

**151. Explain the FailFast iterator and FailSafe iterator along with examples for each.**

A FailFast iterator is an iterator that throws a **ConcurrentModificationException** if it detects that the underlying collection has been modified while the iterator is being used. This is the default behavior of iterators in the Java Collections Framework. For example, the iterator for a HashMap is FailFast.

**Example:**

**Java**

```
// Java Program to demonstrate FailFast iterator
import java.io.*;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
class GFG {
    public static void main(String[] args)
    {
        HashMap<Integer, String> map = new HashMap<>;
        map.put(1, "one");
        map.put(2, "two");
        Iterator<Map.Entry<Integer, String>> iterator
            = map.entrySet().iterator();
        while (iterator.hasNext()) {
            Map.Entry<Integer, String> entry
                = iterator.next();
            // this will throw a
            // ConcurrentModificationException
            if (entry.getKey() == 1) {
                map.remove(1);
            }
        }
    }
}
```

## Output:

```
Exception in thread "main"
java.util.ConcurrentModificationException
```

A FailSafe iterator does not throw a **ConcurrentModificationException** if the underlying collection is modified while the iterator is being used. Alternatively, it creates a snapshot of the collection at the time the iterator is created and iterates over the snapshot. For example, the iterator for a ConcurrentHashMap is FailSafe.

## Example:

### Java

```
// Java Program to demonstrate FailSafe
import java.io.*;
import java.util.Iterator;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;
class GFG {
    public static void main(String[] args)
    {
        ConcurrentHashMap<Integer, String> map
            = new ConcurrentHashMap<>();
        map.put(1, "one");
        map.put(2, "two");
        Iterator<Map.Entry<Integer, String>> iterator
            = map.entrySet().iterator();
        while (iterator.hasNext()) {
            Map.Entry<Integer, String> entry =
                iterator.next();
            // this will not throw an exception
            if (entry.getKey() == 1) {
                map.remove(1);
            }
        }
    }
}
```

## 152. What is Exception Handling?

An **Exception** is an Event that interrupts the normal flow of the program and requires special processing. During the execution of a program, errors and unplanned occurrences can be dealt with by using the Java Exception Handling mechanism. Below are some reasons why Exceptions occur in Java:

- Device failure
- Loss of Network Connection
- Code Errors
- Opening an Unavailable file
- Invalid User Input
- Physical Limitations (out of disk memory)

## 153. How many types of exceptions can occur in a Java program?



There are generally two types of exceptions in Java:

- **Built-in Exceptions:** Built-in exceptions in Java are provided by the Java Libraries. These exceptions can be further divided into two subcategories i.e., checked and unchecked Exceptions. Below are some of the built-in exceptions in Java:
  - `ArrayIndexOutOfBoundsException`
  - `ClassNotFoundException`
  - `FileNotFoundException`

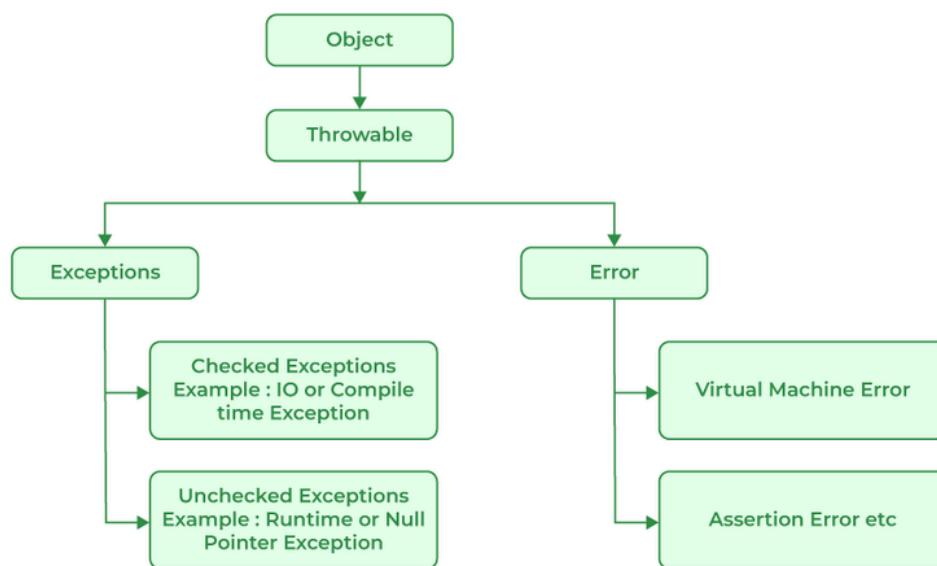
- IOException
- NullPointerException
- ArithmeticException
- InterruptedException
- RuntimeException
- **User-Defined Exceptions:** User-defined exceptions are defined by the programmers themselves to handle some specific situations or errors which are not covered by built-in exceptions. To define user-defined exceptions a new class that extends the appropriate exception class must be defined. User-defined Exceptions in Java are used when the built-in exceptions are in Java.

#### 154. Difference between an Error and an Exception.

Errors	Exceptions
Recovering from Errors is not possible.	Recover from exceptions by either using a try-catch block or throwing exceptions back to the caller.
Errors are all unchecked types in Java.	It includes both checked as well as unchecked types that occur.
Errors are mostly caused by the environment in which the program is running.	The program is mostly responsible for causing exceptions.
Errors can occur at compile time as well as run time. Compile Time: Syntax Error, Run Time: Logical Error.	All exceptions occur at runtime but checked exceptions are known to the compiler while unchecked are not.
They are defined in java.lang.Error package.	They are defined in java.lang.Exception package

Errors	Exceptions
<p><b>Examples:</b>          java.lang.StackOverflowError,          java.lang.OutOfMemoryError</p>	<p><b>Examples:</b> Checked Exceptions:          SQLException, IOException          Unchecked Exceptions:          ArrayIndexOutOfBoundsException,          NullPointerException,          ArithmeticException.</p>

### 155. Explain the hierarchy of Java Exception classes.



All exception and error types in Java are subclasses of the class `Throwable`, which is the base class of the hierarchy. This class is then used for exceptional conditions that user programs should catch. `NullPointerException` is an example of such an exception. Another branch, `Error` is used by the Java run-time system to indicate errors having to do with the JRE. `StackOverflowError` is an example of one of such error.

### 156. Explain Runtime Exceptions.

Runtime Exceptions are exceptions that occur during the execution of a code, as opposed to compile-time exceptions that occur during

compilation. Runtime exceptions are unchecked exceptions, as they aren't accounted for by the JVM.

### Examples of runtime exceptions in Java include:

- `NullPointerException`: This occurs when an application attempts to use a null object reference.
- `ArrayIndexOutOfBoundsException`: This occurs when an application attempts to access an array index that is out of bounds.
- `ArithmaticException`: This occurs when an application attempts to divide by zero.
- `IllegalArgumentException`: This occurs when a method is passed on an illegal or inappropriate argument.

Unlike checked exceptions, runtime exceptions do not require a declaration in the throws clause or capture in a try-catch block. However, handling runtime exceptions is advisable in order to provide meaningful error messages and prevent a system crash. Because runtime exceptions provide more specific information about the problem than checked exceptions, they enable developers to detect and correct programming errors more easily and quickly.

#### 157. What is `NullPointerException`?

It is a type of run-time exception that is thrown when the program attempts to use an object reference that has a null value. The main use of `NullPointerException` is to indicate that no value is assigned to a reference variable, also it is used for implementing data structures like linked lists and trees.

#### 158. When is the `ArrayStoreException` thrown?

`ArrayStoreException` is thrown when an attempt is made to store the wrong type of object in an array of objects.

#### Example:

Java

```
    // Java Program to implement  
    // ArrayStoreException  
    public class GFG {  
        public static void main(String args[])  
        {  
            // Since Double class extends Number class  
            // only Double type numbers  
            // can be stored in this array  
            Number[] a = new Double[2];  
            // Trying to store an integer value  
            // in this Double type array  
            a[0] = new Integer(4);  
        }  
    }
```

### Example:

```
Exception in thread "main" java.lang.ArrayStoreException:  
java.lang.Integer  
at GFG.main(GFG.java:6)
```

## 159. What is the difference between Checked Exception and Unchecked Exception?

### Checked Exception:

Checked Exceptions are the exceptions that are checked during compile time of a program. In a program, if some code within a method throws a checked exception, then the method must either handle the exception or must specify the exception using the throws keyword.

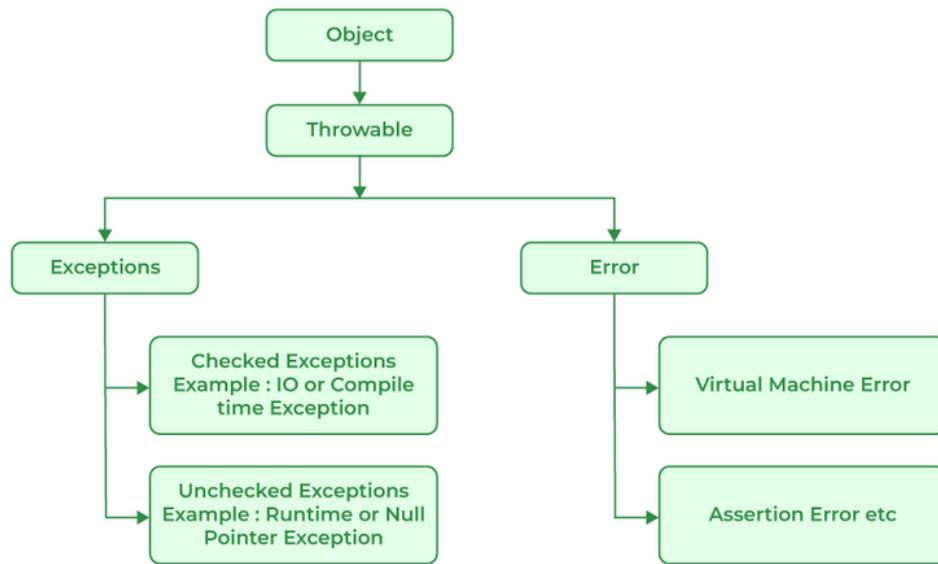
Checked exceptions are of two types:

- Fully checked exceptions: all its child classes are also checked, like IOException, and InterruptedException.
- Partially checked exceptions: some of its child classes are unchecked, like an Exception.

### Unchecked Exception:

Unchecked are the exceptions that are not checked at compile time of a program. Exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked.

## 160. What is the base class for Error and Exception?



Error is an illegal operation performed by the user which causes abnormality in the program. Exceptions are the unexpected events or conditions that comes while running the program, exception disrupts the normal flow of the program's instructions.

Errors and Exceptions both have a common parent class which is `java.lang.Throwable` class.

## 161. Is it necessary that each try block must be followed by a catch block?

No, It is not necessary to use catch block after try block in Java as we can create another combination with finally block. Finally is the block which runs despite the fact that the exception is thrown or not.

## 162. What is exception propagation?

Exception propagation is a process in which the exception is dropped from top to the bottom of the stack. If not caught once, the

exception again drops down to the previous method, and so on until it gets caught or until it reaches the very bottom of the call stack.

### 163. What will happen if you put `System.exit(0)` on the try or catch block? Will finally block execute?

`System.exit(int)` has the capability to throw `SecurityException`. So, if in case of security, the exception is thrown then finally block will be executed otherwise JVM will be closed while calling `System.exit(0)` because of which finally block will not be executed.

### 164. What do you understand by Object Cloning and how do you achieve it in Java?

It is the process of creating an exact copy of any object. In order to support this, a java class has to implement the `Cloneable` interface of `java.lang` package and override the `clone()` method provided by the `Object` class the syntax of which is:

Protected Object clone() throws CloneNotSupportedException{ return (Object)super.clone();}In case the `Cloneable` interface is not implemented and just the method is overridden, it results in `CloneNotSupportedException` in Java.

### 165. How do exceptions affect the program if it doesn't handle them?

Exceptions are responsible for abruptly terminating the running of the program while executing and the code written after the exception occurs is not executed.

### 166. What is the use of the final keyword?

The final keyword is used to make functions non-virtual. By default, all the functions are virtual so to make it non-virtual we use the final keyword.

## 167. What purpose do the keywords final, finally, and finalize fulfill?

### i). final:

final is a keyword used with the variable, method, or class so that they can't be overridden.

Example:

Java

```
// Java Program to use final
// keyword
import java.io.*;
// Driver Class
class GFG {
    // Main function
    public static void main(String[] args)
    {
        final int x = 100;
        x = 50;
    }
}
```

Output:

```
./GFG.java:6: error: cannot assign a value to final variable x
    x=50;
               ^
1 error
```

### ii). finally

finally is a block of code used with “try-catch” in exception handling. Code written in finally block runs despite the fact exception is thrown or not.

Example:

## Java

```
// Java Program to implement finally
import java.io.*;
// Driver class
class GFG {
    // Main function
    public static void main(String[] args)
    {
        int x = 10;
        // try block
        try {
            System.out.println("Try block");
        }
        // finally block
        finally {
            System.out.println(
                "Always runs even without exceptions");
        }
    }
}
```

## Output

Try block  
Always runs even without exceptions

### iii). finalize

It is a method that is called just before deleting/destructing the objects which are eligible for Garbage collection to perform clean-up activity.

Example:

**Java**

```

    /*package whatever // do not write package name here */
    import java.io.*;
    class GFG {
        public static void main(String[] args)
        {
            System.out.println("Main function running");
            System.gc();
        }
        // Here overriding finalize method
        public void finalize()
        {
            System.out.println("finalize method overridden");
        }
    }

```

**Output**

Main function running

**168. What is the difference between this() and super() in Java?**

this( )	super( )
It represents the current instance of the class.	It represents the current instance of the parent class.
Calls the default constructor of the	Calls the default constructor of the

this( )	super( )
same class.	base class.
Access the methods of the same class.	Access the methods of the parent class.
Points current class instance.	Points the superclass instance.

## 169. What is multitasking?

Multitasking in Java refers to a program's capacity to carry out several tasks at once. Threads, which are quick operations contained within a single program, can do this. Executing numerous things at once is known as multitasking.

Example:

**Java**

```

// Java program for multitasking
import java.io.*;
public class MyThread extends Thread {
    public void run()
    {
        // Code to be executed in this thread
        for (int i = 0; i < 10; i++) {
            System.out.println(
                "Thread " + Thread.currentThread().getId()
                + ":" + i);
        }
    }
}
public class GFG {
    public static void main(String[] args)
    {
        MyThread thread1 = new MyThread();
        MyThread thread2 = new MyThread();
        // Start the threads
        thread1.start();
        thread2.start();
    }
}

```

## 170. What do you mean by a Multithreaded program?

Multithreaded programs in Java contain threads that run concurrently instead of running sequentially. A computer can use its resources more efficiently by combining multiple tasks at once. Any program with multithreading allows more than one user to simultaneously use the program without running multiple copies. A multithreaded program is designed to run multiple processes at the same time which can improve the performance of a program and allows the program to utilize multiple processors and improves the overall throughput.

## 171. What are the advantages of multithreading?

There are multiple advantages of using multithreading which are as follows:

- **Responsiveness:** User Responsiveness increases because multithreading interactive application allows running code even when the section is blocked or executes a lengthy process.
- **Resource Sharing:** The process can perform message passing and shared memory because of multithreading.
- **Economy:** We are able to share memory because of which the processes are economical.
- **Scalability:** Multithreading on multiple CPU machines increases parallelism.
- **Better Communication:** Thread synchronization functions improves inter-process communication.
- **Utilization of multiprocessor architecture**
- **Minimized system resource use**

## 172. What are the two ways in which Thread can be created?

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of the CPU. In

general, threads are small, lightweight processes with separate paths of execution. These threads use shared memory, but they act independently, thus if any one thread fails it does not affect the other threads. There are two ways to create a thread:

- By extending the Thread class
- By implementing a Runnable interface.

#### By extending the Thread class

We create a class that extends the **java.lang.Thread class**. This class overrides the run() method available in the Thread class. A thread begins its life inside run() method.

#### Syntax:

```
public class MyThread extends Thread {  
    public void run() {  
        // thread code goes here  
    }  
}
```

#### By implementing the Runnable interface

We create a new class that implements **java.lang.Runnable** interface and override run() method. Then we instantiate a Thread object and call the start() method on this object.

#### Syntax:

```
public class MyRunnable implements Runnable {  
    public void run() {  
        // thread code goes here  
    }  
}
```

## 173. What is a thread?

Threads in Java are subprocess with lightweight with the smallest unit of processes and also has separate paths of execution. These threads use shared memory but they act independently hence if there is an exception in threads that do not affect the working of other threads despite them sharing the same memory. A thread has its own program counter, execution stack, and local variables, but it shares the same memory space with other threads in the same process. Java provides built-in support for multithreading through the **Runnable interface** and the **Thread class**.

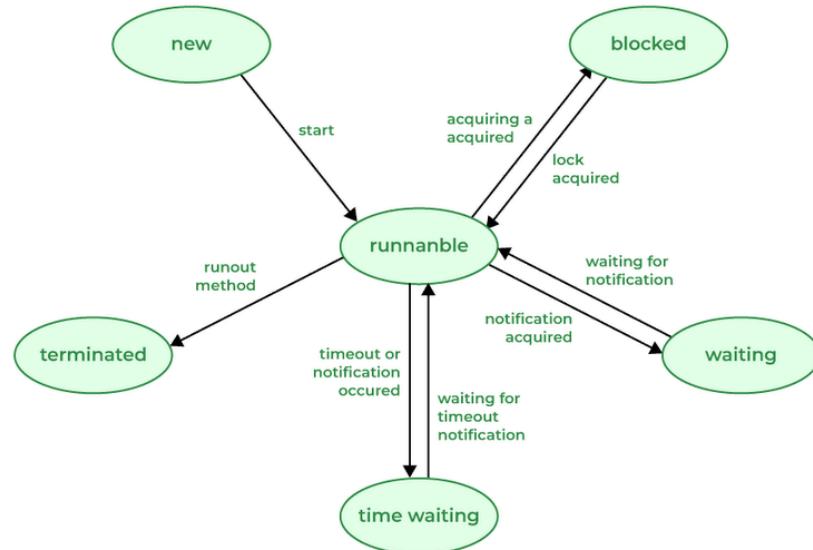
### 174. Differentiate between process and thread?

A process and a thread are both units of execution in a computer system, but they are different in several ways:

Process	Thread
A process is a program in execution.	A thread is a single sequence of instructions within a process.
The process takes more time to terminate.	The thread takes less time to terminate.
The process takes more time for context switching.	The thread takes less time for context switching.
The process is less efficient in terms of communication.	Thread is more efficient in terms of communication.
The process is isolated.	Threads share memory.
The process has its own Process Control Block, Stack, and Address Space.	Thread has Parents' PCB, its own Thread Control Block, and Stack and common Address space.

Process	Thread
The process does not share data with each other.	Threads share data with each other.

### 175. Describe the life cycle of the thread?



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

1. **New:** The thread has been created but has not yet started.
2. **Runnable:** The thread is running, executing its task, or is ready to run if there are no other higher-priority threads.
3. **Blocked:** The thread is temporarily suspended, waiting for a resource or an event.
4. **Waiting:** The thread is waiting for another thread to perform a task or for a specified amount of time to elapse.
5. **Terminated:** The thread has completed its task or been terminated by another thread.

### 176. Explain suspend() method under the Thread class.

The `suspend()` method of the `Thread` class in Java temporarily suspends the execution of a thread. When a thread is suspended it

goes into a blocked state and it would not be scheduled by the operating system which means that it will not be able to execute its task until it is resumed. There are more safer and flexible alternatives to the suspend() methods in the modern java programming language. This method does not return any value.

### Syntax:

```
public final void suspend();
```

### Example:

#### Java

```
// Java program to show thread suspend() method
import java.io.*;
class MyThread extends Thread {
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println(" Running thread : " + i);
            try {
                Thread.sleep(1000);
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
class GFG {
    public static void main(String[] args)
    {
        MyThread t1 = new MyThread();
        t1.start();
        try {
            Thread.sleep(3000);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
        // suspend the execution of the thread
        t1.suspend();
        System.out.println("Suspended thread ");
        try {
            Thread.sleep(3000);
        }
        catch (InterruptedException e) {
```

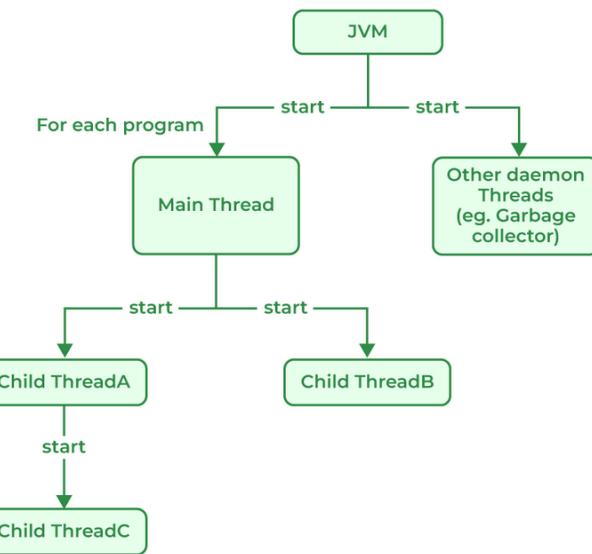
```
        e.printStackTrace();
    }
    // resume the execution of the thread
    t1.resume();
    System.out.println("Resumed thread");
}
}
```

### Output:

```
Thread running: 0
Thread running: 1
Thread running: 2
Suspended thread
Resumed thread
Thread running: 3
Thread running: 4
Thread running: 5
Thread running: 6
Thread running: 7
Thread running: 8
Thread running: 9
```

### 177. Explain the main thread under Thread class execution.

Java provides built-in support for multithreaded programming. The main thread is considered the parent thread of all the other threads that are created during the program execution. The main thread is automatically created when the program starts running. This thread executes the main method of the program. It is responsible for executing the main logic of the Java program as well as handling the user input operations. The main thread serves as the base thread from which all other child threads are spawned.



## 178. What is a daemon thread?

A daemon thread in Java is a low-priority thread that is used to perform background operations or tasks which are used to perform continuously. such as Garbage collection, Signal dispatches, Action listeners, etc. Daemon threads in Java have lower priority than user threads, which means they can only execute when no user threads are running. Daemon threads in Java are useful features that are required for background tasks that do not require explicit shutdown or finalization. It allows more efficient use of system resource and are used to simplify resources and can simplify long-running tasks.

## 179. What are the ways in which a thread can enter the waiting state?

Thread is a lightweight process that runs concurrently with the other thread inside a single process. Each thread can execute a different task and share the resources within a single process. Thread in Java can enter the waiting state in many different ways:

- Sleep() method Call:** The `sleep()` method is used to pause the execution of the thread for a specific amount of time. While the thread is paused it goes into the waiting state.
- Wait() method:** This method is used to wait a thread until the other thread signals it to wake up. Thread goes into the waiting state

until it receives a notification from another thread.

- **Join() method:** Join() method can be used to wait for thread to finish the execution. Calling thread goes into the waiting state until the target thread is completed.
- **Waiting for I/O operations:** If the thread is waiting for Input/Output operation to complete, it goes into the waiting state until the operation is finished.
- **Synchronization Issues:** If there are any synchronization issues in a multi-threaded application, threads may go into the waiting state until the synchronization issues are resolved.

## 180. How does multi-threading take place on a computer with a single CPU?

Java uses a technique called time-sharing, commonly referred to as time-slicing, to implement multi-threading on computers with a single CPU. The appearance of parallel execution is created by the CPU switching between active threads. The operating system is in charge of allocating CPU time to each thread sequentially and scheduling the threads.

In order to stop threads from interacting with one another and creating race situations or other issues, Java has a number of ways to govern the behavior of threads, including synchronization and locking. It is feasible to create multi-threaded programmers that operate correctly and effectively on a machine with a single CPU by regulating the interaction between threads and making sure that crucial code parts are synchronized. In contrast to running the same program on a computer with multiple CPUs or cores, multi-threading on a single CPU can only give the appearance of parallelism, and actual performance gains may be modest. The operating system divides the CPU time that is available when numerous threads are running on a single CPU into small time slices and gives each thread a time slice to execute. Rapid switching between the threads by the operating system creates the appearance of parallel execution. The switching

between threads appears to be immediate because the time slices are often very tiny, on the order of milliseconds or microseconds.

## Java Interview Questions For Experienced

### 181. What are the different types of Thread Priorities in Java? And what is the default priority of a thread assigned by JVM?

Priorities in threads is a concept where every thread is having a priority which in layman's language one can say every object is having priority here which is represented by numbers ranging from 1 to 10. There are different types of thread properties in Java mentioned below:

- MIN\_PRIORITY
- MAX\_PRIORITY
- NORM\_PRIORITY

By default, the thread is assigned NORM\_PRIORITY.

### 182. Why Garbage Collection is necessary in Java?

For Java, Garbage collection is necessary to avoid memory leaks which can cause the program to crash and become unstable. There is no way to avoid garbage collection in Java. Unlike C++, Garbage collection in Java helps programmers to focus on the development of the application instead of managing memory resources and worrying about memory leakage. Java Virtual Machine (JVM) automatically manages the memory periodically by running a garbage collector which frees up the unused memory in the application. Garbage collection makes Java memory efficient because it removes unreferenced objects from the heap memory.

### 183. What is the drawback of Garbage Collection?

Apart from many advantages, Garbage Collector has certain drawbacks mentioned below:

1. The main drawback to Garbage collection is that it can cause pauses in an application's execution as it works to clear the memory which slows down the performance of the application.
2. The Process of Garbage collection is non-deterministic which makes it difficult to predict when garbage collection occurs which causes unpredictable behavior in applications. For Example, if we write any program then it is hard for programmers to decide if the issue is caused by garbage collection or by any other factors in the program.
3. Garbage collection can also increase memory usage if the program creates and discards a lot of short-lived objects.

#### 184. Explain the difference between a minor, major, and full garbage collection.

The Java Virtual Machine (JVM) removes objects that are no longer in use using a garbage collector which periodically checks and removes these objects. There are different types of garbage collection in the JVM, each with different characteristics and performance implications. The main types of garbage collection are:

- **Minor garbage collection:** Also known as young generation garbage collection, this type of garbage collection is used to collect and reclaim memory that is used by short-lived objects (objects that are quickly created and discarded).
- **Major garbage collection:** Also known as old-generation garbage collection, this type of garbage collection is used to collect and reclaim memory that is used by long-lived objects (objects that survive multiple minor garbage collections and are promoted to the old generation).
- **Full garbage collection:** During full garbage collection, memories from all generations are collected and reclaimed, including memories of young and old. A full garbage collection normally takes longer to complete than a minor or major garbage collection which causes that app to pause temporarily.

## 185. How will you identify major and minor garbage collections in Java?

Major garbage collection works on the survivor space and Minor garbage collection works on the Eden space to perform a mark-and-sweep routine. And we can identify both of them based on the output where the minor collection prints “GC”, whereas the major collection prints “Full GC” for the case where the garbage collection logging is enabled with “-XX:PrintGCDetails” or “verbose:gc”.

## 186. What is a memory leak, and how does it affect garbage collection?

In Java Memory leaks can be caused by a variety of factors, such as not closing resources properly, holding onto object references longer than necessary, or creating too many objects unnecessarily. There are situations in which garbage collector does not collect objects because there is a reference to those objects. In these situations where the application creates lots of objects and does not use them and every object has some valid references, a Garbage collector in Java cannot destroy the objects. These useless objects which do not provide any value to the program are known as Memory leaks. Memory leaks can impact garbage collection negatively by preventing the garbage collector from reclaiming unused memory. This behavior will lead to slow performance or sometimes system failure. In a program, it is important to avoid memory leaks by managing resources and object references properly.

**Example:**

Java

```
// Java Program to demonstrate memory Leaks
import java.io.*;
import java.util.Vector;
class GFG {
    public static void main(String[] args)
    {
        Vector a = new Vector(21312312);
```

```
Vector b = new Vector(2147412344);
Vector c = new Vector(219944);
System.out.println("Memory Leak in Java");
}
```

## Output:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap
space
at java.base/java.util.Vector.<init>(Vector.java:142)
at java.base/java.util.Vector.<init>(Vector.java:155)
at GFG.main(GFG.java:9)
```

## 187. Name some classes present in `java.util.regex` package.

Regular Expressions or Regex in Java is an API used for searching and manipulating of strings in Java. It creates String patterns that can extract the data needed from the strings or can generalize a pattern.

There are 3 Classes present in `java.util.regex` mentioned below:

- Pattern Class: Can define patterns
- Matcher Class: Can perform match operations on text using patterns
- PatternSyntaxException Class: Can indicate a syntax error in a regular expression pattern.

Also, apart from the 3 classes package consists of a single interface `MatchResult Interface` which can be used for representing the result of a match operation.

## 188. Write a regular expression to validate a password. A password must start with an alphabet and followed by alphanumeric characters; Its length must be in between 8 to 20.

```
regex = “^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&-+=()])(?=\\S+$.){8, 20}$”
```

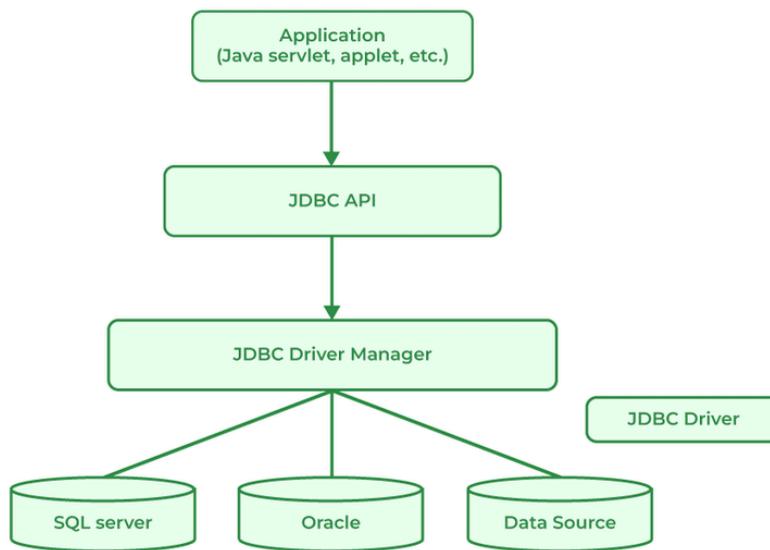
### Explanation:

- ^ used for starting character of the string.
- (?=.\*[0-9]) used for a digit must occur at least once.
- (?=.\*[a-z]) used for a lowercase alphabet must occur at least once.
- (?=.\*[A-Z]) used for an upper case alphabet that must occur at least once in the substring.
- (?=.\*[@#\$%^&-+=()]) used for a special character that must occur at least once.
- (?=\\S+\$) white spaces don't allow in the entire string.
- .{8, 20} used for at least 8 characters and at most 20 characters.
- \$ used for the end of the string.

### 189. What is JDBC?

JDBC standard API is used to link Java applications and relational databases. It provides a collection of classes and interfaces that let programmers to use the Java programming language to communicate with the database. The classes and interface of JDBC allow the application to send requests which are made by users to the specified database. There are generally four components of JDBC by which it interacts with the database:

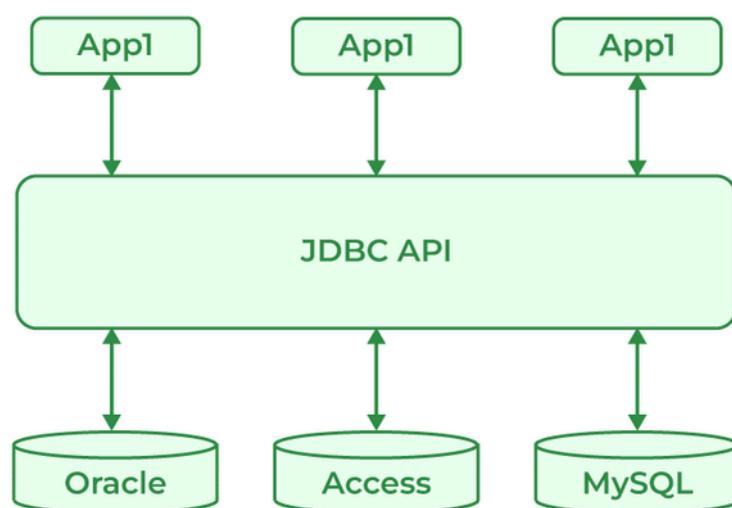
- JDBC API
- JDBC Driver manager
- JDBC Test Suite
- JDBC-ODBC Bridge Drivers



## 190. What is JDBC Driver?

**JDBC Driver** is a software component that is used to enable a Java application to interact with the database. JDBC provides the implementation of the JDBC API for a specific database management system, which allows it to connect the database, execute SQL statements and retrieve data. There are four types of JDBC drivers:

- JDBC-ODBC Bridge driver
- Native-API driver
- Network Protocol driver
- Thin driver



## 191. What are the steps to connect to the database in Java?

There are certain steps to connect the database and Java Program as mentioned below:

- Import the Packages
- Load the drivers using the `forName()` method
- Register the drivers using `DriverManager`
- Establish a connection using the `Connection` class object
- Create a statement
- Execute the query
- Close the connections

## 192. What are the JDBC API components?

JDBC API components provide various methods and interfaces for easy communication with the databases also it provides packages like java Se and java EE which provides the capability of write once run anywhere (WORA).

Syntax:

```
java.sql.*;
```

## 193. What is JDBC Connection interface?

Java database connectivity interface (JDBC) is a software component that allows Java applications to interact with databases. To enhance the connection, JDBC requires drivers for each database.

## 194. What does the JDBC ResultSet interface?

JDBC `ResultSet` interface is used to store the data from the database and use it in our Java Program. We can also use `ResultSet` to update the data using `updateXXX()` methods. `ResultSet` object points the cursor before the first row of the result data. Using the `next()` method, we can iterate through the `ResultSet`.

## 195. What is the JDBC Rowset?

A JDBC RowSet provides a way to store the data in tabular form. RowSet is an interface in java that can be used within the java.sql package. The connection between the RowSet object and the data source is maintained throughout its life cycle. RowSets are classified into five categories based on implementation mentioned below:

1. JdbcRowSet
2. CachedRowSet
3. WebRowSet
4. FilteredRowSet
5. JoinRowSet

## 196. What is the role of the JDBC DriverManager class?

JDBC DriverManager class acts as an interface for users and Drivers. It is used in many ways as mentioned below:

- It is used to create a connection between a Java application and the database.
- Helps to keep track of the drivers that are available.
- It can help to establish a connection between a database and the appropriate drivers.
- It contains all the methods that can register and deregister the database driver classes.
- DriverManager.registerDriver() method can maintain the list of Driver classes that have registered themselves.

# Java Difference Interview Questions

## 197. Differentiate between Iterable and Iterator.

Iterable	Iterator
Iterable provides a way to iterate over a sequence of elements.	Iterator helps in iterating over a collection of elements sequentially.
<code>iterator()</code> method returns an Iterator.	<code>hasNext()</code> and <code>next()</code> methods are required.
<code>remove()</code> method is optional.	<code>remove()</code> method is required in the iterator.
Examples are <b>List</b> , <b>Queue</b> , and <b>Set</b> .	Examples are <b>ListIterator</b> , <b>Enumeration</b> , and <b>ArrayIterator</b> .

### 198. Differentiate between List and Set.

List	Set
Ordered	Unordered
List allows duplicates.	Set does not allow duplicate values.
List is accessed by index.	Set is accessed by hashcode.
Multiple null elements can be stored.	Null element can store only once.
Examples are <b>ArrayList</b> , <b>LinkedList</b> , etc.	Examples are <b>HashSet</b> and <b>TreeSet</b> . <b>LinkedHashSet</b> etc.

### 199. Differentiate between List and Map.

List	Map
List interface allows duplicate elements.	Map does not allow duplicate elements.
The list maintains insertion order.	Map does not maintain insertion order.
Multiple null elements can be stored.	The map allows a single null key at most and any number of null values.
The list provides get() method to get the element at a specified index.	The map does not provide a get method to get the elements at a specified index.
List is Implemented by ArrayList, etc.	Map is Implemented by HashMap, TreeMap, LinkedHashMap

## 200. Differentiate between Queue and Stack.

Queue	Stack
Queue data structure is used to store elements, and is used to perform operations like enqueue, dequeue from back or end of the queue.	Stack data structure is used to store elements, and is used to perform operations like push, pop from top of the stack.
Queue data structure Implements FIFO order.	Stack data structure Implements LIFO order.

Queue	Stack
Insertion and deletion in queues take place from the opposite ends of the list. Deletion takes place from the front of the list and insertion takes place at the rear of the list.	Insertion and deletion in stacks take place only from one end of the list called the top.
Insert operation is called enqueue operation.	Insert operation is called Push operation.
Queue is generally used to solve problems related to sequential processing.	Stack is generally used to solve problems related to recursion.

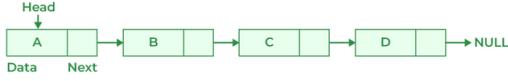
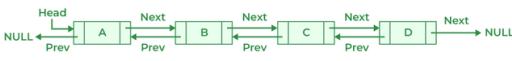
## 201. Differentiate between PriorityQueue and TreeSet.

Priority Queue	TreeSet
It uses Queue as an underlying data structure.	It uses a Set as an underlying data structure.
This data structure allows duplicate elements	This data structure does not allow duplicate elements
Priority Queue is Implemented by PriorityQueue class.	TreeSet is implemented by TreeSet class.
PriorityQueue comes in JDK 1.5.	TreeSet comes in JDK 1.4.
PriorityQueue<Integer> pq = new	TreeSet<Integer> ts = new

Priority Queue	TreeSet
PriorityQueue<>();	TreeSet<>();

## 202. Differentiate between the Singly Linked List and Doubly Linked List.

Singly Linked List	Doubly Linked List
Singly Linked List contain only two segments i.e, Data and Link.	Doubly Linked List contains three segments i.e, Data, and two pointers.
Traversal in a singly linked list is possible in only a forward direction.	Traversal in a doubly linked list is only possible in both directions forward as well as backward.
It uses less memory as every single node has only one pointer.	It requires more memory than a singly linked list as each node has two pointers.
Easy to use and insert nodes at the beginning of the list.	Slightly more complex to use and easy to insert at the end of the list.
The time complexity of insertion and deletion is O(n).	The time complexity of insertion and deletion is O(1).

Singly Linked List	Doubly Linked List
	

### 203. Differentiate between Failfast and Failsafe.

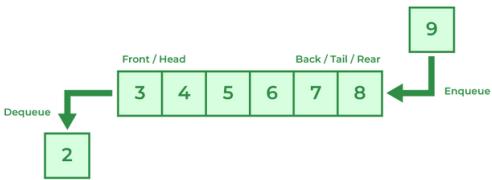
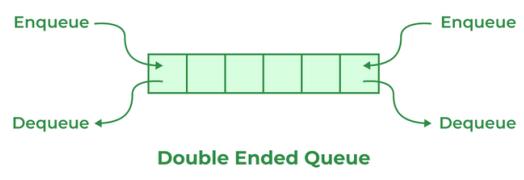
FailFast	FailSafe
Failsafe fails immediately when it detects concurrent modification during the time of iteration.	Failsafe continues to iterate over the original collection and also creates a copy to modify.
Failfast is generally used in single-threaded environments.	Failsafe is used in multithreaded environments.
Failfast does not allow any modification while iteration.	Failsafe allows modification during the time of iteration.
Failfast is fast compared to failsafe as it does not involve the copying of the collection.	Failsafe is generally slow compared to failfast.

FailFast	FailSafe
FailFast throws <b>ConcurrentModificationException</b> if the collection is modified during iteration.	FailSafe does not throw any exception but instead, it creates a copy of the collection to iterate.

## 204. Differentiate between HashMap and TreeMap.

HashMap	TreeMap
HashMap uses a hashtable in order to store key-value pairs.	TreeMap uses Red-black trees to store key-value pair.
HashMap does not maintain any specific order for key-value pairs.	TreeMap maintains a natural ordering based on the keys.
Order of iteration is not guaranteed in the hashmap.	Iteration is of sorted order based on keys.
Hashmaps are faster for retrieval compared to TreeMap.	Retrieval in TreeMap is slower as it uses tree traversal to find keys.
HashMap is implemented by using an Array of linked list.	TreeMap is implemented using a Red-black Tree.
HashMap uses the equals() method of the Object class to compare keys.	TreeMap uses compareTo() method to compare keys.

## 205. Differentiate between Queue and Deque.

Queue	Deque
<p>The queue is a linear Data structure that is used to store a collection of elements.</p>	<p>Deque also known as a Double-ended queue is also a linear data structure that stores a collection of elements with operations to remove and add from both ends.</p>
<p>Elements in the queue can only be inserted at the end of the data structure.</p>	<p>Elements can be inserted from both ends of the data structure.</p>
<p>Queue can be implemented using Array or Linked List.</p>	<p>Dequeue can be implemented using Circular Array or Doubly Linked List.</p>
<p>Queues are generally used to implement a waiting list or task queue.</p>	<p>Deque is used to implement a stack or dequeuing elements from both ends.</p>
	 <p style="text-align: center;"><b>Double Ended Queue</b></p>

## 206. Differentiate between HashSet and TreeSet.

HashSet	TreeSet
HashSet is unordered.	TreeSet is based on natural ordering.
HashSet allows null elements.	TreeSet does not allow null elements.
HashSet is Implemented by the HashSet class.	TreeSet is Implemented by TreeSet class.
<code>HashSet&lt;String&gt; hs = new HashSet&lt;&gt;();</code>	<code>TreeSet&lt;String&gt; ts = new TreeSet&lt;&gt;();</code>

## Java Interview Questions – FAQs

### Q1. What is a Java Developer's salary in India?

*According to various resources, The average salary of a Java Backend Developer is more than **14 lakhs per annum which is 30% higher than any other developer role**. Here you can also check our latest course on [Java Backend Development!](#)*

### Q2. What does Java Developer do?

*A Java developer writes code, designs software solutions, and builds applications using the Java programming language. They collaborate with teams, solve problems, and ensure code quality for efficient and reliable software development.*

### Q3. What are the essential skills required for a Java developer?

A Java developer should have a strong understanding of core Java concepts such as object-oriented programming, data types, control structures, and exception handling. Additionally, knowledge of frameworks like Spring, Hibernate, and web development technologies like Servlets and JSP is beneficial. Other than Technical Skills Problem-solving, debugging, and critical thinking skills are also highly valued.

### Q4. How can I prepare for a Java interview?

To prepare for a Java interview, start by reviewing fundamental Java concepts and practice coding exercises. Study common interview questions related to core Java, data structures, algorithms, and multithreading from [\*\*GeeksforGeeks Interview Section\*\*](#). Additionally, brush up on design patterns, database connectivity, and web development frameworks. Practising coding challenges on platforms like [\*\*GeekforGeeks Practice Portal\*\*](#) can also be helpful.

### Q5. How can I stand out in a Java interview?

To stand out in a Java interview, demonstrate a deep understanding of Java concepts and practical applications. Showcase your problem-solving skills by explaining your approach to complex scenarios and providing efficient solutions. Additionally, highlight any relevant projects or contributions you've made to the Java community. Showing enthusiasm, good

*communication, and a willingness to learn can also leave a positive impression.*

Unlock the Power of Placement Preparation!

Feeling lost in OS, DBMS, CN, SQL, and DSA chaos? Our [Complete Interview Preparation](#) Course is the ultimate guide to conquer placements. Trusted by over 100,000+ geeks, this course is your roadmap to interview triumph.

Ready to dive in? Explore our Free Demo Content and join our [Complete Interview Preparation](#) course.

47

[Suggest improvement](#)[Previous](#)[Next](#)

[Top 50+ Python Interview Questions](#) [Java Collections Interview Questions and Answers \(Latest 2024\)](#) [and Answers](#)

[Share your thoughts in the comments](#)[Add Your Comment](#)

## Similar Reads

[Java Collections Interview Questions and Answers](#)[Top 20 Java Multithreading Interview Questions & Answers](#)[Top 50 TCP/IP interview questions and answers](#)[jQuery Interview Questions and Answers | Set-3](#)[Top 50 IP addressing interview questions and answers](#)[Top 50 Flutter Interview Questions and Answers \(2023\)](#)

[Top 25 Android Interview Questions and Answers For Experienced](#)[Top 40 Flask Interview Questions and Answers](#)[Merge Sort Interview Questions and Answers](#)[Sales Interview Questions and Answers](#)**GeeksforGeeks**

**Article Tags :** [interview-preparation](#), [interview-questions](#), [Java-Interview](#), [Java](#)

**Practice Tags :** [Java](#)



## Company

[About Us](#)  
[Legal](#)  
[Careers](#)  
[In Media](#)  
[Contact Us](#)  
[Advertise with us](#)  
[GFG Corporate Solution](#)  
[Placement Training Program](#)

## Explore

[Job-A-Thon Hiring Challenge](#)  
[Hack-A-Thon](#)  
[GfG Weekly Contest](#)  
[Offline Classes \(Delhi/NCR\)](#)  
[DSA in JAVA/C++](#)  
[Master System Design](#)  
[Master CP](#)  
[GeeksforGeeks Videos](#)  
[Geeks Community](#)

## Languages

[Python](#)  
[Java](#)  
[C++](#)  
[PHP](#)  
[GoLang](#)  
[SQL](#)  
[R Language](#)  
[Android Tutorial](#)

## DSA

[Data Structures](#)  
[Algorithms](#)  
[DSA for Beginners](#)  
[Basic DSA Problems](#)  
[DSA Roadmap](#)  
[DSA Interview Questions](#)  
[Competitive Programming](#)

## Data Science & ML

[Data Science With Python](#)  
[Data Science For Beginner](#)  
[Machine Learning Tutorial](#)  
[ML Maths](#)  
[Data Visualisation Tutorial](#)  
[Pandas Tutorial](#)  
[NumPy Tutorial](#)  
[NLP Tutorial](#)  
[Deep Learning Tutorial](#)

## Web Technologies

[HTML](#)  
[CSS](#)  
[JavaScript](#)  
[TypeScript](#)  
[ReactJS](#)  
[NextJS](#)  
[NodeJs](#)  
[Bootstrap](#)  
[Tailwind CSS](#)

## Python Tutorial

[Python Programming Examples](#)  
[Django Tutorial](#)  
[Python Projects](#)  
[Python Tkinter](#)  
[Web Scraping](#)  
[OpenCV Tutorial](#)

## Computer Science

[GATE CS Notes](#)  
[Operating Systems](#)  
[Computer Network](#)  
[Database Management System](#)  
[Software Engineering](#)  
[Digital Logic Design](#)

Python Interview Question

Engineering Maths

**DevOps**

Git  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

**System Design**

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOAD  
System Design Bootcamp  
Interview Questions

**School Subjects**

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar

**Commerce**

Accountancy  
Business Studies  
Economics  
Management  
HR Management  
Finance  
Income Tax

**UPSC Study Material**

Polity Notes  
Geography Notes  
History Notes  
Science and Technology Notes  
Economy Notes  
Ethics Notes  
Previous Year Papers

**Preparation Corner**

Company-Wise Recruitment Process  
Resume Templates  
Aptitude Preparation  
Puzzles  
Company-Wise Preparation  
Companies  
Colleges

**Competitive Exams**

JEE Advanced  
UGC NET  
SSC CGL  
SBI PO  
SBI Clerk  
IBPS PO  
IBPS Clerk

**More Tutorials**

Software Development  
Software Testing  
Product Management  
Project Management  
Linux  
Excel  
All Cheat Sheets

**Free Online Tools**

Typing Test  
Image Editor  
Code Formatters  
Code Converters  
Currency Converter  
Random Number Generator  
Random Password Generator

**Write & Earn**

Write an Article  
Improve an Article  
Pick Topics to Write  
Share your Experiences  
Internships





Create your own newsletter

Start your own discussion with a newsletter or blog post. Share your thought leadership with every new edition.

[Try it out](#)



# Top 20 C Coding Interview Questions and Answers (2024)



Sushil kumar kushwaha

Campus Ambassador @ Externsclub Pvt Ltd |

Full Stack Developer @ OctaNet & B.Tech...

5 articles

[+ Follow](#)

February 4, 2024

[Open Immersive Reader](#)

## List of 20 C Coding Interview Questions and Answer

Here is a list of 20 C coding interview questions and answers:

### 1. Find the largest number among the three numbers.

- C

```
// C Program to find
```

```
// Largest of three numbers
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int a = 1, b = 2, c = 3;
```

```
// condition for a is greatest
```

```
if (a > b && a > c)
```

```
printf("%d", a);
```

```
// condition for b is greatest
```

```
else if (b > a && b > c)
```

```
printf("%d", b);
```

```
// remaining conditions
```

```
// c is greatest
```

```
else
```

```
printf("%d", c);
```

```
return 0;
```

```
}
```

### Output

3

## 2. Write a Program to check whether a number is prime or not.

```
// C Program for
```

```
// Checking value is
```

```
// Prime or not
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int N = 91;

int flag = 0;

// Iterate from 2 to N/2

for (int i = 2; i <= N / 2; i++) {

    // If n is divisible by any number between 2 and

    // n/2, it is not prime

    if (N % i == 0) {

        flag = 1;

        break;

    }

}

if (flag == 0)

printf("Not a Prime Number");

else

printf("Is a Prime Number");

return 0;
```

### Output

Is a Prime Number

### 3. Write a C program to calculate Compound Interest.

```
// C program to calculate Compound Interest

#include <stdio.h>
```

// For using pow function we must

```
// include math.h

#include <math.h>

// Driver code

int main()

{

    // Principal amount

    double principal = 2300;

    // Annual rate of interest

    double rate = 7;

    // Time

    double time = 4;

    // Calculating compound Interest

    double amount

        = principal * ((pow((1 + rate / 100), time)));

    double CI = amount - principal;

    printf("Compound Interest is : %lf", CI);

    return 0;

}
```

**Output**

```
Compound Interest is : 714.830823
```

**4. Write a Program in C to Swap the values of two variables without using any extra variable.**

```
// C Program to
```

```
// Swap two numbers
```

```
// No Extra Space
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int x = 10;
```

```
    int y = 20;
```

```
    printf("x: %d , y: %d\n", x, y);
```

```
// Code to swap 'x' and 'y'
```

```
    x = x + y;
```

```
    y = x - y;
```

```
    x = x - y;
```

```
    printf("x: %d , y: %d\n", x, y);
```

```
return 0;
```

```
}
```

### Output

```
x: 10 , y: 20
x: 20 , y: 10
```

### 5. Write a Program to Replace all 0's with 1's in a Number.

```
// C Program for
```

```
// Replacing 0 to 1
```

```
#include <math.h>
```

```
#include <stdio.h>
```

```
int main()
{
    int N = 102301;

    int ans = 0;
    int i = 0;

    while (N != 0) {
        // Condition to change value

        if (N % 10 == 0)
            ans = ans + 1 * pow(10, i);

        else
            ans = ans + (N % 10) * pow(10, i);
    }
}
```

```
N = N / 10;
```

```
i++;
```

```
}
```

```
printf("%d", ans);
```

```
return 0;
}
```

**Output:**

```
112311
```

**6. Write a Program to convert the binary number into a decimal number.**

```
// C Program for converting
```

```
// binary to decimal
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N = 11011;
```

```
// Initializing base value a to 1
```

```
    int a = 1;
```

```
    int ans = 0;
```

```
    while (N != 0) {
```

```
        ans = ans + (N % 10) * a;
```

```
        N = N / 10;
```

```
        a = a * 2;
```

```
}
```

```
    printf("%d", ans);
```

```
    return 0;
```

```
}
```

### Output

27

## 7. Write a Program to check if the year is a leap year or not.

```
// C Program to check
```

```
// Year is leap year or not
```

```
#include <stdio.h>
```

```
// Function Declaration to check leap year
```

```
void leap_year(int year)
```

```
{
```

```
// If a year is multiple of 400, then leap year
```

```
if (year % 400 == 0)
```

```
printf("%d is a leap year.\n", year);
```

```
// If a year is multiple of 100, then not a leap year
```

```
else if (year % 100 == 0)
```

```
printf("%d is not a leap year.\n", year);
```

```
// If a year is multiple of 4, then leap year
```

```
else if (year % 4 == 0)
```

```
printf("%d is a leap year.\n", year);
```

```
// Not leap year
```

```
else
```

```
printf("%d is not a leap year.\n", year);
```

```
}
```

```
int main()
```

```
{
```

```
leap_year(2000);
```

```
leap_year(2002);
```

```
leap_year(2008);
```

```
return 0;
```

```
}
```

## Output

```
2000 is a leap year.
```

```
2002 is not a leap year.
```

2008 is a leap year.

## 8. Write a program to Factorial of a Number.

```
// C Program to calculate

// Factorial of a number

#include <stdio.h>

// Calculating factorial using iteration

void factorial_iteration(int N)

{

    unsigned long long int ans = 1;

    for (int i = 1; i <= N; i++) {

        ans = ans * i;

    }

    printf("Factorial of %d is %lld\n", N, ans);

}

// Calculating factorial using recursion

int factorial(int N)

{

    if (N == 0)

        return 1;

    // Recursive call

    return N * factorial(N - 1);

}

int main()

{
```

```
int n;  
  
n = 13;  
  
factorial_iteration(n);  
  
  
n = 9;  
  
printf("Factorial of %d using recursion:%d\n", n,  
factorial(n));  
  
  
return 0;  
}
```

### Output

```
Factorial of 13 is 6227020800  
Factorial of 9 using recursion:362880
```

## 9. Write a Program to Check if a number is an Armstrong number or not.

```
// C program to check if number  
// is Armstrong number or not  
  
#include <stdio.h>  
  
  
// Function to calculate x raised to the power y  
  
int power(int x, unsigned int y)  
{  
    if (y == 0)  
        return 1;  
    if (y % 2 == 0)  
        return power(x, y / 2) * power(x, y / 2);  
  
    return x * power(x, y / 2) * power(x, y / 2);  
}
```

```
// Function to calculate order of the number

int order(int n)

{

    int res = 0;

    while (n) {

        res++;

        n = n / 10;

    }

    return res;

}
```

```
// Function to check whether the given number is

// Armstrong number or not

int isArmstrong(int x)

{

    // Calling order function

    int n = order(x);

    int temp = x, sum = 0;

    while (temp) {

        int r = temp % 10;

        sum += power(r, n);

        temp = temp / 10;

    }

    // If satisfies Armstrong condition

    if (sum == x)

        return 1;

    else

        return 0;

}
```

```
// Driver Program

int main()

{

    int x = 120;

    if (isArmstrong(x) == 1)

        printf("True\n");

    else

        printf("False\n");


    x = 1634;

    if (isArmstrong(x) == 1)

        printf("True\n");

    else

        printf("False\n");


    return 0;
}
```

### Output

```
False
True
```

## 10. Write a program to Find all the roots of a quadratic equation in C.

```
// C program to find roots

// of a quadratic equation

#include <math.h>

#include <stdio.h>

#include <stdlib.h>
```

```
// Prints roots of quadratic equation ax2 + bx + c
```

```
void find_roots(int a, int b, int c)
```

```
{
```

```
// If a is 0, then equation is not quadratic, but
```

```
// linear
```

```
if (a == 0) {
```

```
    printf("Invalid");
```

```
    return;
```

```
}
```

```
int d = (b * b) - (4 * a * c);
```

```
double sqrt_val = sqrt(abs(d));
```

```
if (d > 0) {
```

```
    printf("Roots are real and different \n");
```

```
    printf("%f\n%f", (double)(-b + sqrt_val) / (2 * a),
```

```
        (double)(-b - sqrt_val) / (2 * a));
```

```
}
```

```
else if (d == 0) {
```

```
    printf("Roots are real and same \n");
```

```
    printf("%f", -(double)b / (2 * a));
```

```
}
```

```
else // d < 0
```

```
{
```

```
    printf("Roots are complex \n");
```

```
    printf("%f + i%f\n%f - i%f", -(double)b / (2 * a),
```

```
        sqrt_val / (2 * a), -(double)b / (2 * a),
```

```
        sqrt_val / (2 * a));
```

```
}
```

```
}
```

```
// Driver code

int main()

{
    int a = 1, b = -16, c = 1;

    // Function call

    find_roots(a, b, c);

    return 0;
}
```

**Output:**

```
Roots are real and different
15.937254
0.062746
```

**11. Write a Program to reverse a number.**

```
// C Programs to Calculate

// reverse of a number

#include <stdio.h>

// Iterative approach

int reverse_iteration(int N)

{
    int ans = 0;

    while (N != 0) {

        ans = ans * 10 + (N % 10);

        N = N / 10;
    }

    return ans;
```

}

```
// recursive approach
```

```
int reverse(int n, int ans)
```

{

```
if (n == 0)
```

```
return ans;
```

```
ans = ans * 10 + n % 10;
```

```
return reverse(n / 10, ans);
```

}

```
int main()
```

{

```
int N = 15942;
```

```
printf("Initial number:%d\n", N);
```

```
N = reverse_iteration(N);
```

```
printf("%d after reverse using iteration\n", N);
```

```
int ans = 0;
```

```
ans = reverse(N, ans);
```

```
printf("%d after again reverse using recursion", ans);
```

```
return 0;
```

}

## Output

```
Initial number:15942
24951 after reverse using iteration
15942 after again reverse using recursion
```

## 12. Check whether a number is a palindrome.

```
// C Program for

// Checking Palindrome

#include <stdio.h>

// Checking if the number is

// Palindrome number

void check_palindrome(int N)

{

    int T = N;

    int rev = 0; // This variable stored reversed digit

    // Execute a while loop to reverse digits of given

    // number

    while (T != 0) {

        rev = rev * 10 + T % 10;

        T = T / 10;

    }

    // Compare original_number with reversed number

    if (rev == N)

        printf("%d is palindrome\n", N);

    else

        printf("%d is not a palindrome\n", N);

}

int main()

{

    int N = 13431;

    int M = 12345;
```

```
// Function call  
  
check_palindrome(N);  
  
check_palindrome(M);  
  
  
return 0;  
}
```

### Output

```
13431 is palindrome  
12345 is not a palindrome
```

### 13. Write a C Program to check if two numbers are equal without using the bitwise operator.

```
// C Program for checking numbers  
  
// are equal using bitwise operator  
  
#include <stdio.h>  
  
  
int main()  
{  
    int x = 1;  
    int y = 2;  
  
  
    // Using XOR  
  
    // XOR of two equal numbers is 0  
  
    if (!(x ^ y))  
        printf("%d is equal to %d ", x, y);  
  
    else  
  
        printf("%d is not equal to %d ", x, y);  
  
  
    return 0;
```

}

**Output**

1 is not equal to 2

**14. Write a C program to find the GCD of two numbers.**

```
// C program to find GCD of two numbers

#include <math.h>

#include <stdio.h>

// Function to return gcd of a and b

int gcd(int a, int b)

{

    // Find Minimum of a and b

    int result = ((a < b) ? a : b);

    while (result > 0) {

        if (a % result == 0 && b % result == 0) {

            break;

        }

        result--;

    }

    return result; // return gcd of a and b

}

// Driver program to test above function

int main()

{

    int a = 98, b = 56;

    printf("GCD of %d and %d is %d ", a, b, gcd(a, b));

}
```

```
    return 0;
```

```
}
```

### Output

```
GCD of 98 and 56 is 14
```

## 15. Write a C program to find the LCM of two numbers.

```
// C program to find
```

```
// LCM of two numbers
```

```
#include <stdio.h>
```

```
// minimum of two numbers
```

```
int Min(int Num1, int Num2)
```

```
{
```

```
if (Num1 >= Num2)
```

```
return Num2;
```

```
else
```

```
return Num1;
```

```
}
```

```
int LCM(int Num1, int Num2, int K)
```

```
{
```

```
// If either of the two numbers
```

```
// is 1, return their product
```

```
if (Num1 == 1 || Num2 == 1)
```

```
return Num1 * Num2;
```

```
// If both the numbers are equal
```

```
if (Num1 == Num2)
```

```
return Num1;
```

```
// If K is smaller than the
// minimum of the two numbers

if (K <= Min(Num1, Num2)) {

    // Checks if both numbers are
    // divisible by K or not

    if (Num1 % K == 0 && Num2 % K == 0) {

        // Recursively call LCM() function
        return K * LCM(Num1 / K, Num2 / K, 2);

    }

    // Otherwise
    else
        return LCM(Num1, Num2, K + 1);
}

// If K exceeds minimum
else
    return Num1 * Num2;
}

int main()
{
    // Given N & M
    int N = 12, M = 9;

    // Function Call
    int ans = LCM(N, M, 2);
```

```
printf("%d", ans);
```

```
return 0;
```

```
}
```

### Output

36

**16. Write a C Program to find the Maximum and minimum of two numbers without using any loop or condition.**

```
// C Program to check
```

```
// Maximum and Minimum
```

```
// Between two numbers
```

```
// Without any condition or loop
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
int a = 55, b = 23;
```

```
// return maximum among the two numbers
```

```
printf("max = %d\n", ((a + b) + abs(a - b)) / 2);
```

```
// return minimum among the two numbers
```

```
printf("min = %d", ((a + b) - abs(a - b)) / 2);
```

```
return 0;
```

```
}
```

**Output**

```
max = 55  
min = 23
```

**17. Write a Program in C to Print all natural numbers up to N without using a semi-colon.**

- C

```
// C program to print  
// all natural numbers  
// upto N without using semi-colon  
  
#include <stdio.h>  
  
#define N 10  
  
  
int main(int val)  
{  
    if (val <= N && printf("%d ", val) && main(val + 1)) {  
    }  
}
```

**Output**

```
1 2 3 4 5 6 7 8 9 10
```

**18. Write a Program to find the area of a circle.**

- C

```
// C program to find area
```

```
// of circle
```

```
#include <math.h>
```

```
#include <stdio.h>
```

```
#define PI 3.142
```

```
double findArea(int r) { return PI * pow(r, 2); }
```

```
int main()
```

```
{
```

```
printf("Area is %f", findArea(5));
```

```
return 0;
```

```
}
```

### Output

```
Area is 78.550000
```

## 19. Write a Program to create a pyramid pattern using C.

- C

```
// C Program print Pyramid pattern
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int N = 5;
```

```
// Outer Loop for number of rows
```

```
for (int i = 1; i <= N; i++) {
```

```

// inner Loop for space printing

for (int j = 1; j <= N - i; j++)
    printf(" ");

// inner Loop for star printing

for (int j = 1; j < 2 * i; j++)
    printf("*");
    printf("\n");

}

return 0;
}

```

**Output**

```

*
 ***
 *****
 ******
 *****


```

**20. Write a program to form Pascal Triangle using numbers.**

```

1
    1   1
    1   2   1
    1   3   3   1
    1   4   6   4   1

```

- C

```

// C Program to print

// Pascal's Triangle

#include <stdio.h>

```

```
int main()
{
    int n = 5;

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++) {
            printf(" ");
        }

        int x = 1;

        for (int j = 1; j <= i; j++) {
            printf("%d ", x);
            x = x * (i - j) / j;
        }

        printf("\n");
    }

    return 0;
}
```

### Output

```
1
      1   1
      1   2   1
      1   3   3   1
      1   4   6   4   1
```

---

Report this

Published by

Sushil kumar kushwaha

Campus Ambassador @ Externsclub Pvt Ltd | ★ Full Stack Developer @ Octa...



Published • 3mo

5

articles

[+ Follow](#)

## Top 20 C Coding Interview Questions and Answers (2024)

This article on C Coding Interview Questions offers a comprehensive collection of practice questions suitable for both beginners and advanced learners.

List of 20 C Coding Interview Questions and Answer

Here is a list of 20 C coding interview questions and answers:



Like



Comment



Share



11

## Reactions



+3

0 Comments



Add a comment...



## Sushil kumar kushwaha

Campus Ambassador @ Externsclub Pvt Ltd | 🌟 Full Stack Developer @ OctaNet & B.Tech CSE Student | Expertise in C, C++, Python, JavaScript & More | Android Developer | 100k Views on LinkedIn!

[+ Follow](#)

## More from Sushil kumar kushwaha



15+ Web Development Projects With Source Code [2024]

Sushil kumar kushwaha on Li...



100+ JavaScript Projects With Source Code

Sushil kumar kushwaha on Li...



Data Structure and Algorithms

Sushil kumar kushwaha on Li...

[See all 5 articles](#)

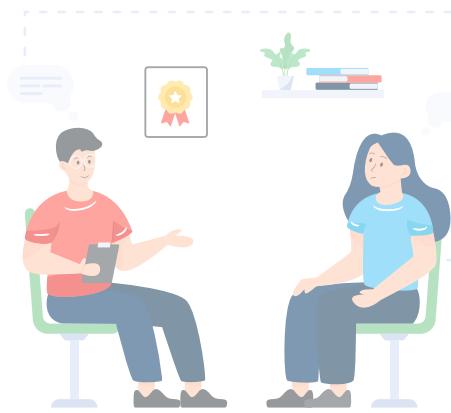




- 3.7. What is a stack?
- 3.8. What are binary trees?
- 3.9. What are binary search trees?
- 3.10. What is object-oriented programming?

#### 4. Coding Interview Questions with Solution

- 4.1. What is the purpose of a loop in programming?
- 4.2. What is a conditional statement?
- 4.3. What is debugging?
- 4.4. What is recursion?
- 4.5. What are the differences between linear and non-linear data structures?
- 4.6. What programming languages do you have experience with?
- 4.7. Describe a time you faced a challenge in a project and how you overcame it.
- 4.8. Walk me through a project you're currently working on.
- 4.9. Give an example of a project where you had to work with a team. How did you learn it?
- 4.10. How do you ensure your code is readable and maintainable?
- 4.11. What are your interests outside of programming?
- 4.12. How do you keep your skills sharp and updated?
- 4.13. How do you collaborate on projects with others?
- 4.14. Tell me about a time when you had to explain a complex concept to a team member.
- 4.15. How do you get started on a new coding project?
- 4.16. Explain the concept of time complexity in algorithms.
- 4.17. What is the significance of version control in software development?
- 4.18. Can you explain the differences between static and dynamic typing?
- 4.19. What is a design pattern in software development?
- 4.20. How do you handle exceptions and errors in your code?



#### Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

#### 5. Conclusion

#### 6. FAQs

- 6.1. What's the purpose of the STAR method in job interviews?
- 6.2. What's the difference between procedural and object-oriented programming?
- 6.3. How does continuous integration benefit software development?
- 6.4. Why is algorithm analysis important in computer science?
- 6.5. Can you provide an example of the Factorial problem?
- 6.6. What's the difference between HTTP and HTTPS protocols?
- 6.7. How do you approach code refactoring in a legacy system?
- 6.8. What are design principles like SOLID in software engineering?
- 6.9. What's the role of load balancing in web architecture?
- 6.10. Why is responsive web design important for user experience?

## Basic Coding Questions

- How do you reverse a string in Python?

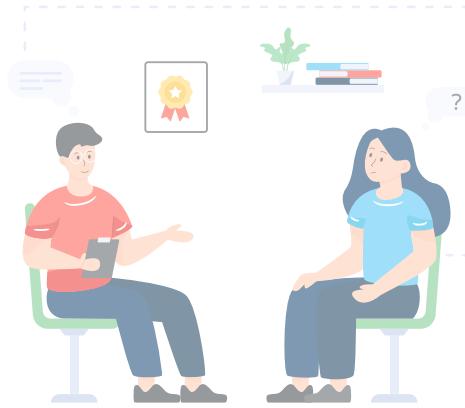


Reversing a string is a common coding task.  
characters and constructing a reversed string.

#### Python Code

```
def reverse_string(input_string):  
  
    return input_string[::-1]
```

#### AI Digital



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

A palindrome is a word or phrase that reads the same backward as forward. You can simply compare it to its reverse to determine if it's a palindrome.

#### Advanced AI & Python

#### Python Code

```
def is_palindrome(input_string):  
  
    return input_string == input_string[::-1]
```

#### Product M

#### UI/UX

## • How can you compute if a character in a string is a numeral?

You can inspect each character within the string to see if it's a numeral.

#### DevOps & Cloud

#### Python Code

```
def count_numerical_digits(input_string):
```

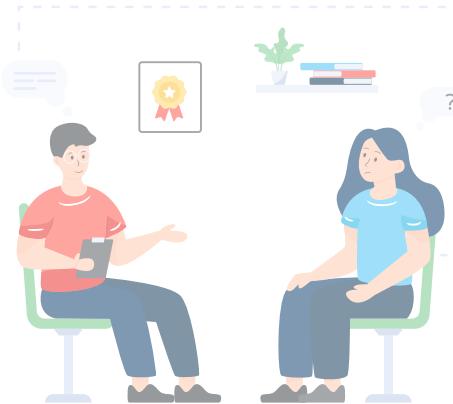
```
    count = 0

    for char in input_string:

        if char.isdigit():

            count += 1

    return count
```



## Recommended Technical Courses

- [Full Stack Development Course](#)
- [Generative AI Course](#)
- [DSA C++ Course](#)
- [Data Analytics Course](#)
- [Python DSA Course](#)
- [DSA Java Course](#)

Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

- **How do you find the occurrences of a particular character in a string?**

You can traverse the string and tally the instances of the character.

### Python Code

```
def count_character_occurrences(input_string, char):
    count = 0

    for c in input_string:

        if c == char:

            count += 1

    return count
```

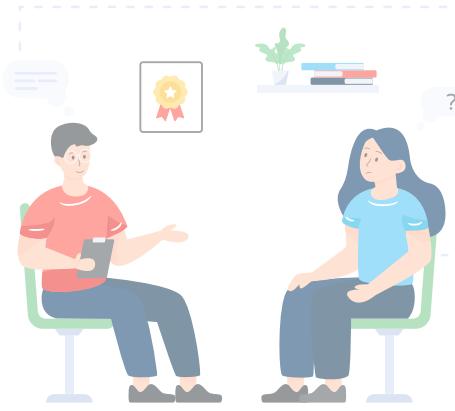


## • How do you find the non-matching characters in two strings?

You can detect different characters in two strings by comparing corresponding characters and noting discrepancies.

### Python Code

```
def find_non_matching_characters(str1, str2):
    non_matching = []
    for i in range(len(str1)):
        if str1[i] != str2[i]:
            non_matching.append(str1[i])
    return non_matching
```



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

## • How do you find out if two strings are anagrams?

Two strings are anagrams if they have the same set of characters in each string and compare them.

### Python Code

```
def are_anagrams(str1, str2):
    return sorted(str1) == sorted(str2)
```

## • How do you calculate the number of consonants in a string?

You can iterate through the string and count the consonants.

### Python Code

```
def count_vowels_and_consonants(input_string):
    vowels = "aeiou"
    consonants = 0
    for char in input_string:
        if char in vowels:
            vowels_count += 1
        else:
            consonants += 1
```



```
num_vowels = 0
```

```
num_consonants = 0
```

```
for char in input_string:
```

```
    if char.isalpha():
```

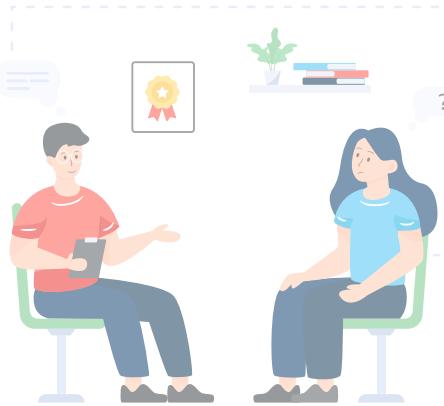
```
        if char.lower() in vowels:
```

```
            num_vowels += 1
```

```
        else:
```

```
            num_consonants += 1
```

```
return num_vowels, num_consonants
```



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

## • How do you total all elements in an array

Iterate through the array, adding up matching elements.

### Python Code

```
def sum_matching_elements(arr, target):
```

```
    total = 0
```

```
    for num in arr:
```

```
        if num == target:
```

```
            total += num
```

```
    return total
```

## • How do you reverse

Reversing an array can be done by swapping the middle of the array.

### Python Code



```
def reverse_array(arr):\n\n    left = 0\n\n    right = len(arr) - 1\n\n    while left < right:\n\n        arr[left], arr[right] = arr[right], arr[left]\n\n        left += 1\n\n        right -= 1
```

- **How do you find the array?**

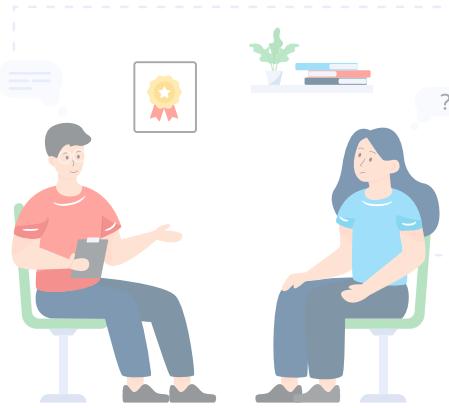
To discover the array's highest value, scan th

#### Python Code

```
def find_max_element(arr):\n\n    if not arr:\n\n        return None\n\n    max_element = arr[0]\n\n    for num in arr:\n\n        if num > max_element:\n\n            max_element = num\n\n    return max_element
```

## Basic Coding Questions

- **How do you sort an array in ascending order?**



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.



You can employ sorting methods like bubble sort in programming languages.

#### Python Code

```
def sort_arrayAscending(arr):  
    arr.sort()
```

### • How do you print a Fibonacci sequence using recursion?

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones. You can print it using recursion.

#### Python Code

```
def print_fibonacci(n, a=0, b=1):  
  
    if n == 0:  
        return  
  
    print(a)  
  
    print_fibonacci(n - 1, b, a + b)
```

### • How do you calculate the sum of two integers?

To calculate the sum of two integers, you can use the addition operator (+).

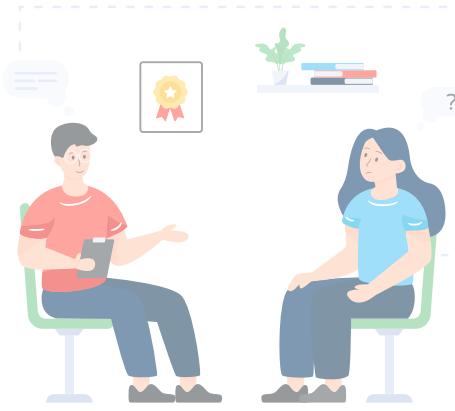
#### Python Code

```
def sum_of_integers(a, b):  
  
    return a + b
```

### • How do you find the average of numbers in a list?

To find the average of numbers in a list, sum all the numbers and divide by the count of numbers.

#### Python Code



Want to Enrol in PW Skills Courses

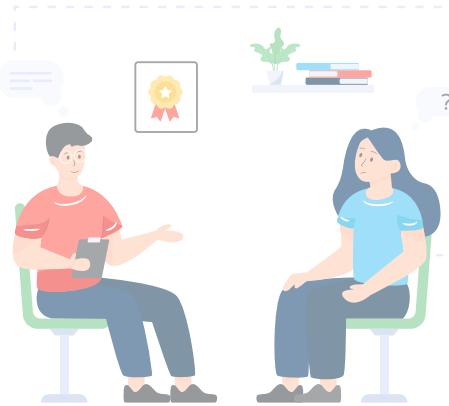
Connect with our experts to get a free counselling & get all your doubt cleared.



```
def average_of_numbers(numbers):\n\n    if not numbers:\n\n        return 0\n\n    return sum(numbers) / len(numbers)
```

## • How do you check if

You can check if an integer is even by dividin



### Python Code

```
def is_even(number):\n\n    return number % 2 == 0
```

## • How do you find the list?

To determine a linked list's midpoint, employ one step, while the other advances by two s

### Python Code

```
def find_middle_element(head):\n\n    slow = head\n\n    fast = head\n\n    while fast is not None and fast.next is not\n\n        slow = slow.next\n\n        fast = fast.next.next\n\n    return slow
```

## • How do you remove

Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

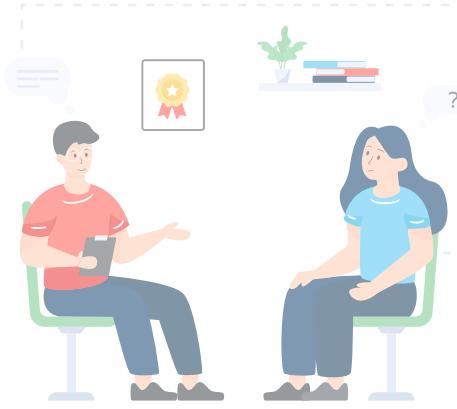


Detecting and removing a loop in a linked list

involves Floyd's Tortoise and Hare algorithm.

### Python Code

```
def detect_and_remove_loop(head):  
  
    # Implementation varies, depending on the  
    # problem statement.  
  
    # You can use Floyd's Tortoise and Hare al-  
    # gorithm to detect the loop.  
  
    # and then remove it.
```



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

## • How do you merge two sorted linked lists?

Merging two sorted linked lists is a common operation in linked list structures.

### Python Code

```
class ListNode:  
  
    def __init__(self, val=0, next=None):  
        self.val = val  
        self.next = next  
  
    def merge_sorted_lists(l1, l2):  
        dummy = ListNode(0)  
        current = dummy  
  
        while l1 is not None and l2 is not None:  
            if l1.val < l2.val:  
                current.next = l1  
                l1 = l1.next  
            else:  
                current.next = l2  
                l2 = l2.next  
  
        current.next = l1 or l2
```



```
current.next = l2
```

```
l2 = l2.next
```

```
current = current.next
```

```
current.next = l1 or l2
```

```
return dummy.next
```

## • How do you implement an element in a sorted array?

Binary search is an efficient algorithm to find an element in a sorted array.

### Python Code

```
def binary_search(arr, target):
```

```
    left, right = 0, len(arr) - 1
```

```
    while left <= right:
```

```
        mid = left + (right - left) // 2
```

```
        if arr[mid] == target:
```

```
            return mid
```

```
        elif arr[mid] < target:
```

```
            left = mid + 1
```

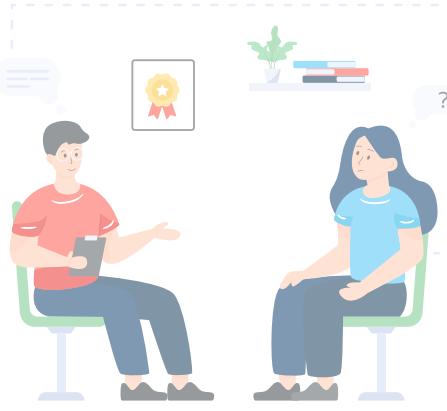
```
        else:
```

```
            right = mid - 1
```

```
    return -1 # Element not found
```

## • How do you print a binary tree in vertical order?

Printing a binary tree in vertical order involves using a dictionary to store nodes at different vertical levels.



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.



## Python Code

```
class TreeNode:

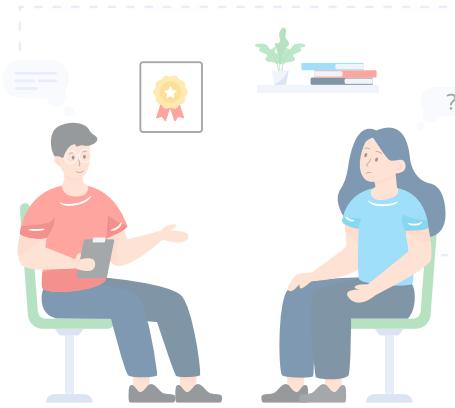
    def __init__(self, val=0, left=None, right=N
                 self.val = val
                 self.left = left
                 self.right = right

    def print_vertical_order(root):

        def vertical_order_traversal(node, distance):
            if node is None:
                return
            if distance in result:
                result[distance].append((level, node.val))
            else:
                result[distance] = [(level, node.val)]
            vertical_order_traversal(node.left, distance + 1)
            vertical_order_traversal(node.right, distance + 1)

        result = {}
        vertical_order_traversal(root, 0, 0, result)

        for distance in sorted(result.keys()):
            for _, val in sorted(result[distance]):
                print(val, end=' ')
            print()
```



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

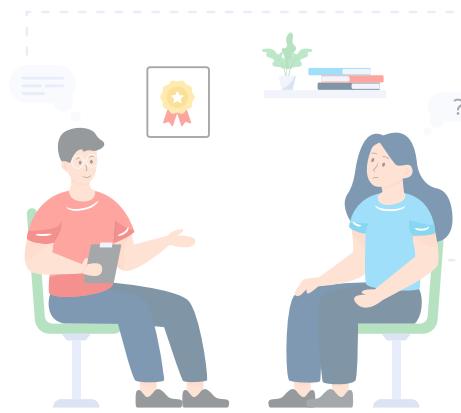
Also Check: [C++ Programming Basics Ever](#)



# Basic Coding Questions

## • What is a data structure?

A data structure organises and stores data in such a way that it can be easily accessed, modified, and used. It defines how data elements relate and what operations are valid on them.



## • What is an array?

An array stores elements, each identified by its index. It's useful for manipulating a group of items of the same kind.

Want to Enrol in PW Skills Courses

## • What is a linked list?

A linked list is a linear structure where each element contains a reference to the next element. It allows for efficient insertion and deletion of elements.

Connect with our experts to get a free counselling & get all your doubt cleared.

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

Arrays have a fixed size, but linked lists can grow or shrink dynamically. Accessing elements in an array is fast, but inserting or deleting elements is slower. In contrast, linked lists offer constant time complexity for both access and modification, with a trade-off in memory usage.

## • What is LIFO (Last-In-First-Out)?

LIFO, a concept in data structures, means the last item added is the first one taken out. It's often linked with stack data structures.

## • What is FIFO (First-In-First-Out)?

FIFO means the first element added is the first one taken out. It's often associated with queue data structures.

## • What is a stack?

A stack is a linear data structure that adheres to the LIFO principle. It enables actions such as pushing to insert an element and popping to remove it.

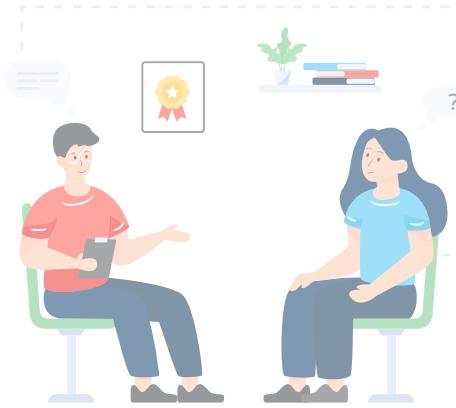


## • What are binary trees?

Binary trees are tree structures with nodes having at most two children, referred to as the left child and the right child.

## • What are binary search trees?

A binary search tree sorts values. The left side contains smaller values, and the right side contains greater values. This setup makes searching much faster.



Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

## Coding Interview Questions

### • What is the purpose of loops in programming?

Loops in programming are used to repeat a set of instructions. They are essential for performing repetitive tasks, iterations, and calculations that involve multiple iterations.

### • What is a conditional statement?

A conditional statement in programming enables a program to make decisions based on the truth or falsehood of a given condition. Common examples include “if” and “else” statements, and “switch” statements.

### • What is debugging?

Debugging means locating and rectifying errors in a program. It involves running the code to pinpoint issues and making necessary changes to ensure proper operation.

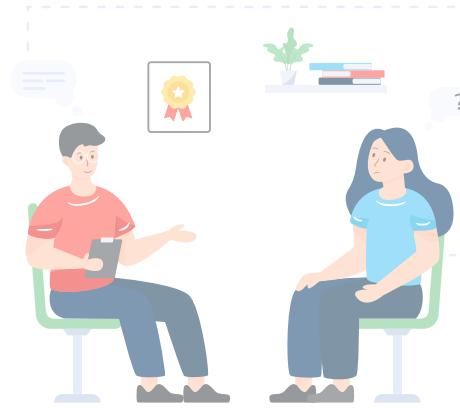
### • What is recursion?

**Recursion**, a programming method, involves a function calling itself repeatedly until it reaches a base case. It is particularly valuable for addressing issues that can be solved by breaking them down into smaller, similar subproblems.



## • What are the differences between linear and non-linear data structures?

Arrays and linked lists hold data one after another in a linear sequence. Linear structures have a simple, sequential connection between elements. Non-linear ones allow complex relationships.



## • What programming experience working on projects do you have?

During coding interviews, you might face questions about your previous experience with various programming languages. Get ready to talk about your proficiency in Python, Java, C++, and others, along with their advantages and drawbacks.

Want to Enrol in PW Skills Courses

Connect with our experts to get a free counselling & get all your doubt cleared.

## • Describe a time you had to work under pressure or manage multiple tasks simultaneously.

Use the STAR method (Situation, Task, Action, Result) to describe a specific challenge, your role, actions taken, and the outcome.

## • Walk me through a project you have recently worked on.

Provide an overview, your role, technologies used, and the challenges faced.

## • Give an example of a time you had to learn a new programming language or technology. How did you approach it?

New Skill Acquisition: Detail the necessity, challenges, resources used, steps taken, difficulties faced, and impact on project success.

## • How do you ensure your code is readable and maintainable for other developers?

Code Quality: Highlight best practices like clear variable names, meaningful comments, modular design, unit testing, and code reviews.



# Top 50 C Coding Interview Questions and Answers (2024)

Last Updated : 22 Jan, 2024

**C** is the most popular programming language developed by Dennis Ritchie at the Bell Laboratories in 1972 to develop the UNIX operating systems. It is a general-purpose and procedural programming language. It is faster than the languages like Java and Python. C is the most used language in top companies such as LinkedIn, Microsoft, Opera, Meta, and NASA because of its performance. To get into these companies and other software companies, you need to master some important C coding questions to crack their C Online Assessment round and coding interview.



*C Coding Interview Questions and Answers*

This article on **C Coding Interview Questions** offers a comprehensive collection of practice questions suitable for both beginners and advanced learners.

## List of 50 C Coding Interview Questions and Answer

Here is a list of 50 C coding interview questions and answers:

1. Find the largest number among the three numbers.

# C

```
// C Program to find
// Largest of three numbers
#include <stdio.h>

int main()
{
    int a = 1, b = 2, c = 3;

    // condition for a is greatest
    if (a > b && a > c)
        printf("%d", a);

    // condition for b is greatest
    else if (b > a && b > c)
        printf("%d", b);

    // remaining conditions
    // c is greatest
    else
        printf("%d", c);

    return 0;
}
```

## Output

3

## 2. Write a Program to check whether a number is prime or not.

---

### C

```
// C Program for
// Checking value is
// Prime or not
#include <stdio.h>

int main()
{
    int N = 91;
    int flag = 0;

    // Iterate from 2 to N/2
    for (int i = 2; i <= N / 2; i++) {

        // If n is divisible by any number between 2 and
        // n/2, it is not prime
        if (N % i == 0) {
            flag = 1;
            break;
        }
    }

    if (flag == 0)
        printf("Not a Prime Number");
    else
        printf("Is a Prime Number");
    return 0;
}
```

## Output

Is a Prime Number

### 3. Write a C program to calculate Compound Interest.

---

## C

```
// C program to calculate Compound Interest
#include <stdio.h>

// For using pow function we must
// include math.h
#include <math.h>

// Driver code
int main()
{
    // Principal amount
    double principal = 2300;

    // Annual rate of interest
    double rate = 7;

    // Time
    double time = 4;

    // Calculating compound Interest
    double amount
        = principal * ((pow((1 + rate / 100), time)));
    double CI = amount - principal;

    printf("Compound Interest is : %lf", CI);
    return 0;
}
```

## Output

Compound Interest is : 714.830823

#### 4. Write a Program in C to Swap the values of two variables without using any extra variable.

---

## C

```
// C Program to
// Swap two numbers
// No Extra Space
#include <stdio.h>

int main()
{
    int x = 10;
    int y = 20;

    printf("x: %d , y: %d\n", x, y);

    // Code to swap 'x' and 'y'
    x = x + y;
    y = x - y;
    x = x - y;

    printf("x: %d , y: %d\n", x, y);

    return 0;
}
```

## Output

```
x: 10 , y: 20
x: 20 , y: 10
```

#### 5. Write a Program to Replace all 0's with 1's in a Number.

---

## C

```
// C Program for
// Replacing 0 to 1
#include <math.h>
```

```

#include <stdio.h>

int main()
{
    int N = 102301;

    int ans = 0;
    int i = 0;
    while (N != 0) {
        // Condition to change value
        if (N % 10 == 0)
            ans = ans + 1 * pow(10, i);
        else
            ans = ans + (N % 10) * pow(10, i);

        N = N / 10;
        i++;
    }

    printf("%d", ans);

    return 0;
}

```

## Output:

**112311**

**6. Write a Program to convert the binary number into a decimal number.**

---

## C

```

// C Program for converting
// binary to decimal
#include <stdio.h>

int main()
{
    int N = 11011;

    // Initializing base value a to 1

```

```

int a = 1;
int ans = 0;
while (N != 0) {
    ans = ans + (N % 10) * a;
    N = N / 10;
    a = a * 2;
}

printf("%d", ans);
return 0;
}

```

## Output

27

## 7. Write a Program to check if the year is a leap year or not.

---

### C

```

// C Program to check
// Year is leap year or not
#include <stdio.h>

// Function Declaration to check leap year
void leap_year(int year)
{
    // If a year is multiple of 400, then leap year
    if (year % 400 == 0)
        printf("%d is a leap year.\n", year);

    // If a year is multiple of 100, then not a leap year
    else if (year % 100 == 0)
        printf("%d is not a leap year.\n", year);

    // If a year is multiple of 4, then leap year
    else if (year % 4 == 0)
        printf("%d is a leap year.\n", year);

    // Not leap year
    else
        printf("%d is not a leap year.\n", year);
}

```

```

int main()
{
    leap_year(2000);
    leap_year(2002);
    leap_year(2008);

    return 0;
}

```

## Output

2000 is a leap year.  
 2002 is not a leap year.  
 2008 is a leap year.

## 8. Write a program to Factorial of a Number.

---

### C

```

// C Program to calculate
// Factorial of a number
#include <stdio.h>

// Calculating factorial using iteration
void factorial_iteration(int N)
{
    unsigned long long int ans = 1;
    for (int i = 1; i <= N; i++) {
        ans = ans * i;
    }

    printf("Factorial of %d is %lld\n", N, ans);
}

// Calculating factorial using recursion
int factorial(int N)
{
    if (N == 0)
        return 1;

    // Recursive call
    return N * factorial(N - 1);
}

```

```

}

int main()
{
    int n;
    n = 13;
    factorial_iteration(n);

    n = 9;
    printf("Factorial of %d using recursion:%d\n", n,
           factorial(n));

    return 0;
}

```

## Output

Factorial of 13 is 6227020800  
 Factorial of 9 using recursion:362880

**9. Write a Program to Check if a number is an Armstrong number or not.**

---

## C

```

// C program to check if number
// is Armstrong number or not
#include <stdio.h>

// Function to calculate x raised to the power y
int power(int x, unsigned int y)
{
    if (y == 0)
        return 1;
    if (y % 2 == 0)
        return power(x, y / 2) * power(x, y / 2);

    return x * power(x, y / 2) * power(x, y / 2);
}

// Function to calculate order of the number
int order(int n)

```

```
{  
    int res = 0;  
    while (n) {  
        res++;  
        n = n / 10;  
    }  
    return res;  
}  
  
// Function to check whether the given number is  
// Armstrong number or not  
int isArmstrong(int x)  
{  
    // Calling order function  
    int n = order(x);  
    int temp = x, sum = 0;  
    while (temp) {  
        int r = temp % 10;  
        sum += power(r, n);  
        temp = temp / 10;  
    }  
  
    // If satisfies Armstrong condition  
    if (sum == x)  
        return 1;  
    else  
        return 0;  
}  
  
// Driver Program  
int main()  
{  
    int x = 120;  
    if (isArmstrong(x) == 1)  
        printf("True\n");  
    else  
        printf("False\n");  
  
    x = 1634;  
    if (isArmstrong(x) == 1)  
        printf("True\n");  
    else  
        printf("False\n");  
  
    return 0;  
}
```

## Output

False  
True

10. Write a program to Find all the roots of a quadratic equation in C.

---

## C

```
// C program to find roots
// of a quadratic equation
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

// Prints roots of quadratic equation ax*x + bx + c
void find_roots(int a, int b, int c)
{
    // If a is 0, then equation is not quadratic, but
    // linear
    if (a == 0) {
        printf("Invalid");
        return;
    }

    int d = (b * b) - (4 * a * c);
    double sqrt_val = sqrt(abs(d));

    if (d > 0) {
        printf("Roots are real and different \n");
        printf("%f\n%f", (double)(-b + sqrt_val) / (2 * a),
               (double)(-b - sqrt_val) / (2 * a));
    }
    else if (d == 0) {
        printf("Roots are real and same \n");
        printf("%f", -(double)b / (2 * a));
    }
    else // d < 0
    {
        printf("Roots are complex \n");
        printf("%f + i%f\n%f - i%f", -(double)b / (2 * a),
               sqrt_val / (2 * a), -(double)b / (2 * a),
               sqrt_val / (2 * a));
    }
}
```

```

    }

// Driver code
int main()
{
    int a = 1, b = -16, c = 1;

    // Function call
    find_roots(a, b, c);
    return 0;
}

```

**Output:**

```

Roots are real and different
15.937254
0.062746

```

**11. Write a Program to reverse a number.****C**

```

// C Programs to Calculate
// reverse of a number
#include <stdio.h>

// Iterative approach
int reverse_iteration(int N)
{
    int ans = 0;
    while (N != 0) {

        ans = ans * 10 + (N % 10);
        N = N / 10;
    }

    return ans;
}

// recursive approach
int reverse(int n, int ans)
{
    if (n == 0)

```

```

    return ans;

    ans = ans * 10 + n % 10;
    return reverse(n / 10, ans);
}

int main()
{
    int N = 15942;
    printf("Initial number:%d\n", N);

    N = reverse_iteration(N);
    printf("%d after reverse using iteration\n", N);

    int ans = 0;
    ans = reverse(N, ans);
    printf("%d after again reverse using recursion", ans);

    return 0;
}

```

## Output

```

Initial number:15942
24951 after reverse using iteration
15942 after again reverse using recursion

```

## 12. Check whether a number is a palindrome.

---

### C

```

// C Program for
// Checking Palindrome
#include <stdio.h>

// Checking if the number is
// Palindrome number
void check_palindrome(int N)
{
    int T = N;
    int rev = 0; // This variable stored reversed digit

    // Execute a while loop to reverse digits of given

```

```

// number
while (T != 0) {
    rev = rev * 10 + T % 10;
    T = T / 10;
}

// Compare original_number with reversed number
if (rev == N)
    printf("%d is palindrome\n", N);
else
    printf("%d is not a palindrome\n", N);
}

int main()
{
    int N = 13431;
    int M = 12345;

    // Function call
    check_palindrome(N);
    check_palindrome(M);

    return 0;
}

```

## Output

13431 is palindrome  
12345 is not a palindrome

**13. Write a C Program to check if two numbers are equal without using the bitwise operator.**

---

## C

```

// C Program for checking numbers
// are equal using bitwise operator
#include <stdio.h>

int main()
{
    int x = 1;

```

```

int y = 2;

// Using XOR
// XOR of two equal numbers is 0
if (!(x ^ y))
    printf("%d is equal to %d ", x, y);
else
    printf("%d is not equal to %d ", x, y);

return 0;
}

```

## Output

1 is not equal to 2

---

### 14. Write a C program to find the GCD of two numbers.

## C

```

// C program to find GCD of two numbers
#include <math.h>
#include <stdio.h>

// Function to return gcd of a and b
int gcd(int a, int b)
{
    // Find Minimum of a and b
    int result = ((a < b) ? a : b);
    while (result > 0) {
        if (a % result == 0 && b % result == 0) {
            break;
        }
        result--;
    }
    return result; // return gcd of a and b
}

// Driver program to test above function
int main()
{
    int a = 98, b = 56;
    printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
}

```

```
    return 0;  
}
```

## Output

```
GCD of 98 and 56 is 14
```

## 15. Write a C program to find the LCM of two numbers.

---

### C

```
// C program to find  
// LCM of two numbers  
#include <stdio.h>  
  
// minimum of two numbers  
int Min(int Num1, int Num2)  
{  
    if (Num1 >= Num2)  
        return Num2;  
    else  
        return Num1;  
}  
  
int LCM(int Num1, int Num2, int K)  
{  
    // If either of the two numbers  
    // is 1, return their product  
    if (Num1 == 1 || Num2 == 1)  
        return Num1 * Num2;  
  
    // If both the numbers are equal  
    if (Num1 == Num2)  
        return Num1;  
  
    // If K is smaller than the  
    // minimum of the two numbers  
    if (K <= Min(Num1, Num2)) {  
  
        // Checks if both numbers are  
        // divisible by K or not  
        if (Num1 % K == 0 && Num2 % K == 0) {
```

```

    // Recursively call LCM() function
    return K * LCM(Num1 / K, Num2 / K, 2);
}

// Otherwise
else
    return LCM(Num1, Num2, K + 1);
}

// If K exceeds minimum
else
    return Num1 * Num2;
}

int main()
{
    // Given N & M
    int N = 12, M = 9;

    // Function Call
    int ans = LCM(N, M, 2);

    printf("%d", ans);

    return 0;
}

```

## Output

36

**16.** Write a C Program to find the Maximum and minimum of two numbers without using any loop or condition.

## C

```

// C Program to check
// Maximum and Minimum
// Between two numbers
// Without any condition or loop
#include <stdio.h>
#include <stdlib.h>

```

```

int main()
{
    int a = 55, b = 23;

    // return maximum among the two numbers
    printf("max = %d\n", ((a + b) + abs(a - b)) / 2);

    // return minimum among the two numbers
    printf("min = %d", ((a + b) - abs(a - b)) / 2);

    return 0;
}

```

## Output

```

max = 55
min = 23

```

**17. Write a Program in C to Print all natural numbers up to N without using a semi-colon.**

---

## C

```

// C program to print
// all natural numbers
// upto N without using semi-colon
#include <stdio.h>
#define N 10

int main(int val)
{
    if (val <= N && printf("%d ", val) && main(val + 1)) {
    }
}

```

## Output

```

1 2 3 4 5 6 7 8 9 10

```

## 18. Write a Program to find the area of a circle.

---

C

```
// C program to find area  
// of circle  
#include <math.h>  
#include <stdio.h>  
#define PI 3.142  
  
double findArea(int r) { return PI * pow(r, 2); }  
  
int main()  
{  
    printf("Area is %f", findArea(5));  
    return 0;  
}
```

Output

Area is 78.550000

---

## 19. Write a Program to create a pyramid pattern using C.

C

```
// C Program print Pyramid pattern  
#include <stdio.h>  
  
int main()  
{  
    int N = 5;  
  
    // Outer Loop for number of rows  
    for (int i = 1; i <= N; i++) {  
  
        // inner Loop for space printing  
        for (int j = 1; j <= N - i; j++)  
            printf(" ");  
  
        // inner Loop for star printing  
        for (int k = 1; k <= i; k++)  
            printf("*");  
  
        printf("\n");  
    }  
}
```

```

    for (int j = 1; j < 2 * i; j++)
        printf("*");
    printf("\n");
}
return 0;
}

```

## Output

```

*
 ***
 *****
 ******
 ******

```

## 20. Write a program to form Pascal Triangle using numbers.

```

      1
     1   1
    1   2   1
   1   3   3   1
  1   4   6   4   1

```

---

## C

```

// C Program to print
// Pascal's Triangle
#include <stdio.h>

int main()
{
    int n = 5;

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++) {
            printf(" ");
        }

```

```

int x = 1;

for (int j = 1; j <= i; j++) {
    printf("%d ", x);
    x = x * (i - j) / j;
}
printf("\n");

return 0;
}

```

## Output

```

1
1   1
1   2   1
1   3   3   1
1   4   6   4   1

```

## 21. Write a Program to return the nth row of Pascal's triangle.

---

### C

```

// C program to return the Nth row of pascal's triangle
#include <stdio.h>

// Print the N-th row of the Pascal's Triangle
void generateNthRow(int N)
{
    // nC0 = 1
    int prev = 1;
    printf("%d", prev);

    for (int i = 1; i <= N; i++) {
        // nCr = (nCr-1 * (n - r + 1)) / r
        int curr = (prev * (N - i + 1)) / i;
        printf(",%d ", curr);
        prev = curr;
    }
}

```

```
int main()
{
    int n = 5;
    generateNthrow(n);
    return 0;
}
```

## Output

1,5 ,10 ,10 ,5 ,1

## 22. Write a program to reverse an Array.

---

### C

```
// C Program to reverse
// An array
#include <stdio.h>

void reverse(int* arr, int n)
{
    // Swapping front and back elements.
    for (int i = 0, j = n - 1; i < j; i++, j--) {
        int ele = arr[i];
        arr[i] = arr[j];
        arr[j] = ele;
    }
}

int main()
{
    int arr[] = { 1, 2, 3, 4, 5 };
    // Function Call
    reverse(arr, 5);

    // reverse array element printing
    for (int i = 0; i < 5; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

## Output

5 4 3 2 1

### 23. Write a program to check the repeating elements in C.

---

## C

```
// C Program for
// checking duplicate
// values in a array
#include <stdio.h>

int Sort(int arr[], int size)
{
    for (int i = 0; i < size - 1; i++) {

        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

// find repeating element
void findRepeating(int arr[], int n)
{
    int count = 0;
    for (int i = 0; i < n; i++) {

        int flag = 0;
        while (i < n - 1 && arr[i] == arr[i + 1]) {
            flag = 1;
            i++;
        }
        if (flag)
            printf("%d ", (arr[i - 1]));
    }
}
```

```

    return;
}

int main()
{
    int arr[] = { 1, 3, 4, 1, 2, 3, 5, 5 };

    int n = sizeof(arr) / sizeof(arr[0]);

    Sort(arr,n);

    findRepeating(arr,n);

    return 0;
}

```

## Output

1 3 5

**24. Write a Program to print the Maximum and Minimum elements in an array.**

---

## C

```

// C Program for calculating
// maximum and minimum element
#include <stdio.h>

void find_small_large(int arr[], int n)
{
    int min, max;

    // assign first element as minimum and maximum
    min = arr[0];
    max = arr[0];

    for (int i = 1; i < n; i++) {

        // finding smallest here
        if (arr[i] < min)

```

```

        min = arr[i]; // finding largest here
    if (arr[i] > max)
        max = arr[i];
}
printf("Maximum: %d and Minimum: %d\n", min, max);
}

int main()
{
    int arr[] = { 15, 14, 35, 2, 11, 83 };
    int len = sizeof(arr) / sizeof(arr[0]);

    // Function call
    find_small_large(arr, len);

    return 0;
}

```

## Output

Smallest: 2 and Largest: 83

**25. Write a Program for the cyclic rotation of an array to k positions.**

---

## C

```

// C program to rotate
// Array by k elements
#include <stdio.h>

// Print array
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
}

// Calculates greatest common divisor
int gcd(int a, int b)
{
    if (b == 0)

```

```

        return a;
    else
        return gcd(b, a % b);
    }

// Rotate array
void Rotate(int arr[], int k, int N)
{
    int i, j, a, temp;
    k = k % N;

    int rotate = gcd(k, N);

    for (i = 0; i < rotate; i++) {

        temp = arr[i];
        j = i;
        while (1) {
            a = j + k;
            if (a >= N)
                a = a - N;
            if (a == i)
                break;
            arr[j] = arr[a];
            j = a;
        }
        arr[j] = temp;
    }
}

int main()
{
    int arr[] = { 1, 2, 3, 4, 5 };

    // Rotating array
    Rotate(arr, 2, 5);

    // Printing array
    printArray(arr, 5);

    return 0;
}

```

## Output

3 4 5 1 2

## 26. Write a Program to sort First half in Ascending order and the Second in Descending order.

---

### C

```
// C Program for Sorting
// First half in Ascending order
// and Second Descending order
#include <stdio.h>

void Sort_asc_desc(int arr[], int n)
{
    int temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    // printing first half in ascending order
    for (int i = 0; i < n / 2; i++)
        printf("%d ", arr[i]);

    // printing second half in descending order
    for (int j = n - 1; j >= n / 2; j--)
        printf("%d ", arr[j]);
}

int main()
{
    int arr[] = { 11, 23, 42, 16, 83, 73, 59 };
    int N = sizeof(arr) / sizeof(arr[0]);

    Sort_asc_desc(arr, N);

    return 0;
}
```

## Output

11 16 23 83 73 59 42

## 27. Write a Program to print sums of all subsets in an array.

---

### C

```
// C Program to print sum of
// all subsets
#include <stdio.h>

// Function to print sum of subset
// Using recursion
void subset_sum(int arr[], int i, int j, int sum)
{
    if (i > j) {
        printf("%d ", sum);
        return;
    }

    subset_sum(arr, i + 1, j, sum + arr[i]);
    subset_sum(arr, i + 1, j, sum);
}

// driver code
int main()
{
    int arr[] = { 1, 2, 3 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // Function calling to print subset sum
    subset_sum(arr, 0, n - 1, 0);
    return 0;
}
```

## Output

6 3 4 1 5 2 3 0

## 28. Write a Program to Find if there is any subarray with a sum equal to 0.

---

### C

```
// C Program to check 0 sum
// subarray possible
#include <stdio.h>

int main()
{
    // array
    int arr[] = { -2, 2, 1, 1, 8 };
    int n = sizeof(arr) / sizeof(arr[0]);

    int flag = 0, sum;

    // Traversing array to check
    for (int i = 0; i < n; i++) {

        for (int j = i; j < n; j++) {
            sum += arr[j];

            if (sum == 0) {
                flag = 1;
                printf(
                    "True subarray with 0 sum is possible");
                break;
            }
        }

        if (flag == 0)
            printf("No such condition");
    }
}
```

### Output

True subarray with 0 sum is possible

## 29. Write a C program to Implement Kadane's Algorithm

# C

```
// C program to implement Kadane's Algorithm
#include <limits.h>
#include <stdio.h>

int main()
{
    int a[] = { -2, -3, 4, -1, -2, 1, 5, -3 };
    int n = sizeof(a) / sizeof(a[0]);

    int max_so_far = INT_MIN, max_ending_here = 0,
        start = 0, end = 0, s = 0;

    for (int i = 0; i < n; i++) {
        max_ending_here += a[i];

        if (max_so_far < max_ending_here) {
            max_so_far = max_ending_here;
            start = s;
            end = i;
        }

        if (max_ending_here < 0) {
            max_ending_here = 0;
            s = i + 1;
        }
    }

    printf("Maximum contiguous sum is %d\n", max_so_far);
    printf("Starting index %d Ending index %d", start, end);

    return 0;
}
```

## Output

Maximum contiguous sum is 7  
 Starting index 2 Ending index 6

## 30. Write a Program to find the transpose of a matrix.

# C

```
#include <stdio.h>

// This function stores transpose of A[][] in B[][]
void transpose(int N, int M, int A[M][N], int B[N][M])
{
    int i, j;
    for (i = 0; i < N; i++)
        for (j = 0; j < M; j++)
            B[i][j] = A[j][i];
}

int main()
{
    int M = 3;
    int N = 4;

    int A[3][4] = { { 1, 1, 1, 1 },
                    { 2, 2, 2, 2 },
                    { 3, 3, 3, 3 } };

    // Note dimensions of B[][]
    int B[N][M], i, j;

    transpose(N, M, A, B);

    printf("Result matrix is \n");
    for (i = 0; i < N; i++) {
        for (j = 0; j < M; j++)
            printf("%d ", B[i][j]);
        printf("\n");
    }

    return 0;
}
```

## Output

```
Result matrix is
1 2 3
1 2 3
1 2 3
```

---

### 31. Write a Program to Rotate a matrix by 90 degrees in the clockwise direction in C.

---

## C

```
// C Program to rotate the array
// By 90 degree in clockwise direction
#include <stdio.h>

void swap(int* a, int* b){
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int n = 4;
    int arr[4][4] = { { 1, 2, 3, 4 },
                      { 5, 6, 7, 8 },
                      { 9, 10, 11, 12 },
                      { 13, 14, 15, 16 } };

    // Print Original Matrix
    printf("Original Matrix:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }

    // Rotate the matrix about the main diagonal
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++)
            swap(&arr[i][j], &arr[j][i]);
    }

    // Rotate the matrix about middle column
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n / 2; j++) {
            swap(&arr[i][j], &arr[i][n - j - 1]);
        }
    }
}
```

```

        }

    // Print the rotated matrix
    printf("Matrix after rotation: \n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
}
}

```

## Output

Original Matrix:

```

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

```

Matrix after rotation:

```

13 9 5 1
14 10 6 2
15 11 7 3
16 12 8 4

```

---

## 32. Write a Program to find the Spiral Traversal of a Matrix in C.

### C

```

// C Program to find Spiral Traversal
// Of a matrix
#include <stdio.h>

int main()
{
    int arr[4][4] = { { 1, 5, 9, 13 },
                     { 2, 6, 10, 14 },
                     { 3, 7, 11, 15 },
                     { 4, 8, 12, 16 } };

```

```
int m = 4, n = 4;
int i, l = 0, right = m - 1, begin = 0, end = n - 1;

while (l <= right && begin <= end) {

    // Print the first row
    // from the remaining rows
    for (i = l; i <= right; ++i) {
        printf("%d ", arr[begin][i]);
    }
    begin++;

    // Print the last column
    // from the remaining columns
    for (i = begin; i <= end; ++i) {
        printf("%d ", arr[i][right]);
    }
    right--;

    // Print the last row from
    // the remaining rows
    if (begin <= end) {
        for (i = right; i >= l; --i) {
            printf("%d ", arr[end][i]);
        }
        end--;
    }

    // Print the first column from
    // the remaining columns
    if (l <= right) {
        for (i = end; i >= begin; --i) {
            printf("%d ", arr[i][l]);
        }
        l++;
    }
}

return 0;
}
```

## Output

1 5 9 13 14 15 16 12 8 4 3 2 6 10 11 7

### 33. Write a program to count the sum of numbers in a string.

---

**C**

```
#include <stdio.h>

int main()
{
    char s[] = "124259";

    int ans = 0;
    // iterate through all the number
    for (int i = 0; s[i] != '\0'; i++) {
        int ele = s[i] - 48;
        if (ele <= 9)
            ans += ele;
    }

    // print sum of the numbers
    printf("%d", ans);

    return 0;
}
```

**Output**

23

---

### 34. Program to calculate the length of the string.

**C**

```
// C Program to calculate
// length of a string
#include <stdio.h>
#include <string.h>

int length(char s[], int i)
{
    if (s[i] == '\0')
```

```

    return 0;

    return length(s, i + 1) + 1;
}

int main()
{
    char s[] = "GeeksforGeeks";

    // Calculating using strlen
    int len = strlen(s);
    printf("length using strlen:%d\n", len);

    // Calculating using iteration
    int i;
    for (i = 0; s[i] != '\0'; i++) {
        continue;
    }
    printf("length using iteration:%d\n", i);

    // Calculating using recursion
    int ans = length(s, 0);
    printf("length using recursion:%d\n", ans);
    return 0;
}

```

## Output

```

length using strlen:13
length using iteration:13
length using recursion:13

```

## 35. Write a program to check string is a palindrome.

---

### C

```

// C implementation to check if a given
// string is palindrome or not
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

bool is_palindrome(char* str, int i, int j)

```

```

{
    if (i >= j) {
        return true;
    }
    if (str[i] != str[j]) {
        return false;
    }
    return is_palindrome(str, i + 1, j - 1);
}

void check_palindrome(char* str)
{
    // Start from leftmost and
    // rightmost corners of str
    int h = 0;
    int flag = 0;
    int l = strlen(str) - 1;

    // Keep comparing characters
    // while they are same
    while (h < l) {
        if (str[l++] != str[h++]) {
            printf("%s is not a palindrome\n", str);
            flag = 1;
            break;
        }
        // will break from here
    }
}

if (flag == 0)
    printf("%s is a palindrome\n", str);
}

int main()
{
    char str[] = { "GeekeeG" };
    char str2[] = { "GeeksforGeeks" };

    check_palindrome(str);

    printf("Checking %s using recursive approach\n", str2);
    bool ans = is_palindrome(str2, 0, strlen(str2)-1);
    if (ans)
        printf("It is Palindrome\n");
    else
        printf("Not a Palindrome\n");

    return 0;
}

```

}

## Output

```
GeekeeG is a palindrome
Checking GeeksforGeeks using recursive approach
Not a Palindrome
```

**36. Write a program to print all permutations of a given string in lexicographically sorted order in C.**

---

## C

```
// C Program to print all permutations of a string in sorted
// order.
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// function two compare two characters a and b
int compare(const void* a, const void* b)
{
    return (*((char*)a) - *((char*)b));
}

// function two swap two characters a and b
void swap(char* a, char* b)
{
    char t = *a;
    *a = *b;
    *b = t;
}

// function finds the index of the smallest character
int findCeil(char str[], char first, int l, int h)
{
    int ceilIndex = l;

    for (int i = l + 1; i <= h; i++)
        if (str[i] > first && str[i] < str[ceilIndex])
            ceilIndex = i;
```

```

    return ceilIndex;
}

// Print all permutations of str in sorted order
void sortedPermutations(char str[])
{
    int size = strlen(str);

    qsort(str, size, sizeof(str[0]), compare);

    int isFinished = 0;
    while (!isFinished) {
        printf("%s \n", str);

        int i;
        for (i = size - 2; i >= 0; --i)
            if (str[i] < str[i + 1])
                break;

        if (i == -1)
            isFinished = 1;
        else {

            int ceilIndex
                = findCeil(str, str[i], i + 1, size - 1);
            swap(&str[i], &str[ceilIndex]);
            qsort(str + i + 1, size - i - 1, sizeof(str[0]),
                  compare);
        }
    }
}

int main()
{
    char str[] = "123";
    sortedPermutations(str);
    return 0;
}

```

## Output

123  
132  
213  
231  
312

321

---

37. Write a program to calculate the Power of a Number using Recursion in C.

## C

```
// C program to calculate the Power of a Number using
// Recursion
#include <stdio.h>

int power(int a, int b)
{
    if (b == 0)
        return 1;

    return power(a, b - 1) * a;
}

int main()
{
    int a = 4, b = 5;

    int ans = power(a, b);

    printf("%d", ans);
    return 0;
}
```

## Output

1024

---

38. Write a Code to print the Fibonacci series using recursion.

## C

```
// C Program to illustrate
```

```
// Fibonacci Series using Recursion
#include <stdio.h>

int fibonacci(int n)
{
    if (n <= 1)
        return n;
    return fibonacci(n - 1) + fibonacci(n - 2);
}

int fibonacci_iteration(int n)
{
    if (n <= 1)
        return 1;

    int arr[n + 1];
    arr[0] = 1;
    arr[1] = 1;

    for (int i = 2; i < n + 1; i++)
        arr[i] = arr[i - 1] + arr[i - 2];

    return arr[n];
}

int main()
{
    int n = 9;
    printf("Fibonacci using recursion of %d:%d\n", n,
           fibonacci(n));

    n = 11;
    printf("Fibonacci using iteration of %d:%d", n,
           fibonacci_iteration(n));
    return 0;
}
```

## Output

Fibonacci using recursion of 9:34  
 Fibonacci using iteration of 11:144

## 39. Write a Program to find the HCF of two Numbers using Recursion.

## C

```
// C program to find
// GCD of two numbers
#include <stdio.h>

// Recursive function to
// Calculate and return gcd of a and b
int gcd(int a, int b)
{
    // Everything divides 0
    if (a == 0)
        return b;
    if (b == 0)
        return a;

    // base case
    if (a == b)
        return a;

    // a is greater
    if (a > b)
        return gcd(a - b, b);
    return gcd(a, b - a);
}

int main()
{
    int a = 192, b = 36;
    printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
    return 0;
}
```

## Output

GCD of 192 and 36 is 12

40. Write a Program in C to reverse a string using recursion.

## C

```
// C program to reverse
// String using recursion
#include <stdio.h>
#include <string.h>

// Using Iteration for reverse
void reverse_iteration(char* str)
{
    int i = 0;
    int j = strlen(str) - 1;

    for (; i < j; i++, j--) {
        char temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}

// Using recursion for reverse
void reverse(char* str)
{
    if (*str) {
        reverse(str + 1);
        printf("%c", *str);
    }
}

int main()
{
    char a[] = "Geeks for Geeks";
    printf("Original string:%s\n", a);

    reverse_iteration(a);
    printf("Reverse the string(iteration):%s\n", a);

    printf("Using recursion for reverse:");
    reverse(a);

    return 0;
}
```

## Output

Original string:Geeks for Geeks

Reverse the string(iteration):skeeG rof skeeG

Using recursion for reverse:Geeks for Geeks

## 41. Write a C Program to search elements in an array.

---

### C

```
// C code to Search elements in array
#include <stdio.h>

int search(int arr[], int N, int x)
{
    int i;

    // iterate through all the element of array
    for (i = 0; i < N; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

int main(void)
{
    int arr[] = { 9, 3, 2, 1, 10, 4 };
    int x = 10;
    int N = sizeof(arr) / sizeof(arr[0]);

    // Function Call
    int result = search(arr, N, x);

    if (result == -1) {
        printf("Element is not present in array");
    }
    else {
        printf("Element is present at index %d", result);
    }

    return 0;
}
```

## Output

```
Element is present at index 4
```

## 42. Write a C Program to search elements in an array using Binary Search.

---

### C

```
// C program to Search element
// in Array using Binary Search
#include <stdio.h>

int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - 1) / 2;

        // If the element is present at the middle
        // itself
        if (arr[mid] == x)
            return mid;

        // If element is smaller than mid, then
        // it can only be present in left subarray
        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);

        // Else the element can only be present
        // in right subarray
        return binarySearch(arr, mid + 1, r, x);
    }

    return -1;
}

int main()
{
    int arr[] = { 11, 14, 19, 23, 40 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 40;
    int result = binarySearch(arr, 0, n - 1, x);
    if (result == -1) {
        printf("Element is not present in array");
    }
}
```

```

    else {
        printf("Element is present at index %d", result);
    }
    return 0;
}

```

## Output

Element is present at index 4

**43. Write a C Program to sort arrays using Bubble, Selection, and Insertion Sort.**

---

## C

```

// C Program to implement
// Sorting Algorithms
#include <stdio.h>

// A function to implement bubble sort
void bubble_sort(int* arr, int n)
{
    for (int j = 0; j < n - 1; j++) {

        // Last j elements are already in place
        for (int i = 0; i < n - j - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
            }
        }
    }

    // A function to implement swaping
    void swap(int* xp, int* yp)
    {
        int temp = *xp;
        *xp = *yp;
        *yp = temp;
    }
}

```

```
// A function to implement selectionSort
void selectionSort(int arr[], int n)
{
    // One by one move boundary of unsorted subarray
    for (int i = 0; i < n - 1; i++) {
        // Find the minimum element in unsorted array
        int min_idx = i;
        for (int j = i + 1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        // Swap the found minimum element
        // with the first element
        if (min_idx != i)
            swap(&arr[min_idx], &arr[i]);
    }
}

void insertionSort(int arr[], int n)
{
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;

        // Move elements of arr that are
        // greater than key, to one position ahead
        // of their current position
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

int main()
{
    int arr1[] = { 9, 4, 3, 11, 1, 5 };
    int arr2[] = { 4, 3, 9, 1, 5, 11 };
    int arr3[] = { 5, 1, 11, 3, 4, 9 };
    int n = 6;

    printf("Non-Sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr1[i]);
    printf("\n");
}
```

```
// sort array
bubble_sort(arr1, n);

// printing array
printf("Sorted array using Bubble sort: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr1[i]);
printf("\n");

printf("Non-Sorted array: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr2[i]);
printf("\n");

// sort array
insertionSort(arr2, n);

// printing array
printf("Sorted array using Insertion Sort: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr2[i]);
printf("\n");

printf("Non-Sorted array: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr3[i]);
printf("\n");

// sort array
selectionSort(arr3, n);

// printing array
printf("Sorted array using Selection Sort: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr3[i]);
printf("\n");

return 0;
}
```

## Output

```
Non-Sorted array: 9 4 3 11 1 5
Sorted array using Bubble sort: 1 3 4 5 9 11
Non-Sorted array: 4 3 9 1 5 11
Sorted array using Insertion Sort: 1 3 4 5 9 11
```

Non-Sorted array: 5 1 11 3 4 9

Sorted array using Selection Sort: 1 3 4 5 9 11

#### 44. Write a C Program to sort arrays using Merge Sort.

---

## C

```
// C program for
// Sorting array
// using Merge Sort
#include <stdio.h>

void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    // create temporary arrays
    int L[n1], R[n2];

    // Copy data to arrays from L[] and R[]
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    // Initial index of first ,second
    // and merged subarray respectively
    i = 0;
    j = 0;
    k = l;

    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    // Copy the remaining elements of L[]
}
```

```
while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
}

// Copy the remaining elements of R[]
while (j < n2) {
    arr[k] = R[j];
    j++;
    k++;
}

void mergeSort(int arr[], int l, int r)
{
    if (l < r) {

        // calculating middle term
        int mid = l + (r - 1) / 2;

        // divide to sort both halves
        mergeSort(arr, l, mid);
        mergeSort(arr, mid + 1, r);

        merge(arr, l, mid, r);
    }
}

int main()
{
    int arr[] = { 23, 9, 13, 15, 6, 7 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // Printing original array
    printf("Given array:");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");

    mergeSort(arr, 0, n - 1);

    // Printing sorted array
    printf("Sorted array :");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");

    return 0;
}
```

}

## Output

```
Given array:23 9 13 15 6 7
Sorted array :6 7 9 13 15 23
```

## 45. Write a C Program to sort arrays using Quick Sort.

---

### C

```
// C Program for
// sorting array using
// Quick sort
#include <stdio.h>

void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

int partition(int array[], int low, int high)
{
    int pivot = array[high];

    int i = (low - 1);

    // compare elements with the pivot
    for (int j = low; j < high; j++) {
        if (array[j] <= pivot) {
            i++;
            swap(&array[i], &array[j]);
        }
    }

    // swap the pivot element with the greater element at i
    swap(&array[i + 1], &array[high]);

    return (i + 1);
}
```

```

void quickSort(int array[], int low, int high)
{
    if (low < high) {
        int pi = partition(array, low, high);
        quickSort(array, low, pi - 1);
        quickSort(array, pi + 1, high);
    }
}

void printArray(int array[], int n)
{
    for (int i = 0; i < n; ++i) {
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main()
{
    int arr[] = { 28, 7, 20, 1, 10, 3, 6 };

    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Unsorted Array:");
    printArray(arr, n);

    quickSort(arr, 0, n - 1);

    printf("Sorted array :");
    printArray(arr, n);

    return 0;
}

```

## Output

```

Unsorted Array:28 7 20 1 10 3 6
Sorted array :1 3 6 7 10 20 28

```

## 46. Write a program to sort an array using pointers.

---

## C

```
// C Program to implement
// sorting using pointers
#include <stdio.h>

// Function to sort the numbers using pointers
void sort(int n, int* ptr)
{
    int i, j;

    // Sort the numbers using pointers
    for (i = 0; i < n; i++) {

        for (j = i + 1; j < n; j++) {

            if (*(ptr + j) < *(ptr + i)) {

                int temp = *(ptr + i);
                *(ptr + i) = *(ptr + j);
                *(ptr + j) = temp;
            }
        }
    }

    // print the numbers
    for (i = 0; i < n; i++)
        printf("%d ", *(ptr + i));
}

// Driver code
int main()
{
    int n = 5;
    int arr[] = { 13, 22, 7, 12, 4 };

    sort(n, arr);

    return 0;
}
```

## Output

4 7 12 13 22

## 47. Write a C program to Store Information about Students Using Structure

---

### C

```
// C Program to Store
// Information about Students
// Using Structure
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Create the student structure
struct Student {
    char* name;
    int roll_number;
    int age;
};

// Driver code
int main()
{
    int n = 3;

    // Create the student's structure variable
    // with n Student's records
    struct Student student[n];

    // Get the students data
    student[0].roll_number = 1;
    student[0].name = "Geeks1";
    student[0].age = 10;

    student[1].roll_number = 2;
    student[1].name = "Geeks2";
    student[1].age = 11;

    student[2].roll_number = 3;
    student[2].name = "Geeks3";
    student[2].age = 13;

    // Printing the Structers
    printf("Student Records:\n\n");
    for (int i = 0; i < n; i++) {
        printf("\tName : %s", student[i].name);
        printf("\tRoll Number : %d",
               student[i].roll_number);
    }
}
```

```

        printf("\tAge : %d\n", student[i].age);
    }

    return 0;
}

```

## Output

Student Records:

Name : Geeks1	Roll Number : 1	Age : 10
Name : Geeks2	Roll Number : 2	Age : 11
Name : Geeks3	Roll Number : 3	Age : 13

---

## 48. Write a C Program To Add Two Complex Numbers Using Structures And Functions.

### C

```

// C program to demonstrate
// addition of complex numbers
#include <stdio.h>

// define a structure for complex number
typedef struct complexNumber {
    int real;
    int img;
} complex;

complex add(complex x, complex y)
{
    // define a new complex number.
    complex add;

    // add similar type together
    add.real = x.real + y.real;
    add.img = x.img + y.img;

    return (add);
}

```

```

int main()
{
    // define three complex type numbers
    complex x, y, sum;

    // first complex number
    x.real = 4;
    x.img = 5;

    // second complex number
    y.real = 7;
    y.img = 11;

    // printing both complex numbers
    printf(" x = %d + %di\n", x.real, x.img);
    printf(" y = %d + %di\n", y.real, y.img);

    // call add(a,b) function and
    // pass complex numbers a & b
    // as an parameter.
    sum = add(x, y);

    // print result
    printf("\n sum = %d + %di", sum.real, sum.img);

    return 0;
}

```

## Output

```

x = 4 + 5i
y = 7 + 11i

sum = 11 + 16i

```

## 49. Write a C Program to add Two Distance Given as Input in Feet and Inches

---

### C

```
// C program for calculating sum of
```

```
// Distance in inches and feet
#include <stdio.h>

// Struct defined for the inch-feet system
struct InchFeet {
    int feet;
    float inch;
};

// Function to find the sum of all N
// set of Inch Feet distances
void findSum(struct InchFeet arr[], int N)
{

    // Variable to store sum
    int feet_sum = 0;
    float inch_sum = 0.0;

    int x;

    // Traverse the InchFeet array
    for (int i = 0; i < N; i++) {

        // Find the total sum of
        // feet and inch
        feet_sum += arr[i].feet;
        inch_sum += arr[i].inch;
    }

    // If inch sum is greater than 11
    // convert it into feet
    // as 1 feet = 12 inch
    if (inch_sum >= 12) {

        // Find integral part of inch_sum
        x = (int)inch_sum;

        // Delete the integral part x
        inch_sum -= x;

        // Add x%12 to inch_sum
        inch_sum += x % 12;

        // Add x/12 to feet_sum
        feet_sum += x / 12;
    }

    // Print the corresponding sum of
    // feet_sum and inch_sum
}
```

```

printf("Feet Sum: %d\n", feet_sum);
printf("Inch Sum: %.2f", inch_sum);
}

int main()
{
    struct InchFeet arr[]
        = { { 11, 5.1 }, { 13, 4.5 }, { 6, 8.1 } };

    int N = sizeof(arr) / sizeof(arr[0]);

    findSum(arr, N);

    return 0;
}

```

## Output

Feet Sum: 31  
 Inch Sum: 5.70

## 50. Write a C program to reverse a linked list iteratively

---

### C

```

// C program to reverse a linked list iteratively
#include <stdio.h>
#include <stdlib.h>

/* Link list node */
struct Node {
    int data;
    struct Node* next;
};

/* Function to reverse the linked list */
static void reverse(struct Node** head_ref)
{
    struct Node* prev = NULL;
    struct Node* current = *head_ref;
    struct Node* next = NULL;
    while (current != NULL) {
        // Store next

```

```
next = current->next;

    // Reverse current node's pointer
    current->next = prev;

    // Move pointers one position ahead.
    prev = current;
    current = next;
}

*head_ref = prev;
}

/* Function to push a node */
void push(struct Node** head_ref, int new_data)
{
    struct Node* new_node
        = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

/* Function to print linked list */
void printList(struct Node* head)
{
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
}

/* Driver code*/
int main()
{
    /* Start with the empty list */
    struct Node* head = NULL;

    push(&head, 10);
    push(&head, 14);
    push(&head, 19);
    push(&head, 25);

    printf("Given linked list\n");
    printList(head);
    reverse(&head);
    printf("\nReversed linked list \n");
    printList(head);
    getchar();
}
```

{}

## Output

Given linked list

25 19 14 10

Reversed linked list

10 14 19 25

## Conclusion

In this C coding interview questions and answers, we've compiled a wide-range of practice questions suitable for individuals at all levels, from beginners to advanced learners. Exploring these questions and their solutions will not only enhance your proficiency in C but also prepare you for a successful coding interview experience.

## C Coding Interview Questions – FAQs

### Q: What are the most common C coding interview questions?

*The most common C coding interview questions are designed to test your knowledge of the following topics:*

- *C syntax and semantics*
- *Data structures and algorithms*
- *Memory management*
- *Pointers*
- *File I/O*

*Some specific examples of common C coding interview questions include:*

- *Reverse a linked list.*
- *Implement a binary search tree.*
- *Write a function to find the maximum element in an array.*

- *Explain the difference between a pointer and an array.*
- *What is the difference between a function declaration and a function definition?*
- *How do you allocate memory on the heap?*
- *How do you free memory that has been allocated on the heap?*
- *What is a dangling pointer?*
- *How do you read and write data to a file?*

## Q. Who can benefit from these C coding interview questions and answers?

*These questions are designed to benefit anyone preparing for a C coding interview. Whether you're a beginner looking to learn the fundamentals or an experienced programmer aiming to enhance your C skills, this resource can assist you in your preparation.*

## Q: How can I use these questions effectively in my interview preparation?

*Start by assessing your current level of expertise in C programming language. Then, you can use these questions to gradually build your skills up and knowledge. Practice solving them on your own, and review the explanations to ensure a thorough understanding.*

Summer-time is here and so is the time to skill-up! More than 5,000 learners have now completed their journey from **basics of DSA to**

advanced level development programs such as Full-Stack, Backend Development, Data Science.

And why go anywhere else when our [DSA to Development: Coding Guide](#) will help you master all this in a few months! Apply now to our [DSA to Development Program](#) and our counsellors will connect with you for further guidance & support.

29

[Suggest improvement](#)[Next](#)

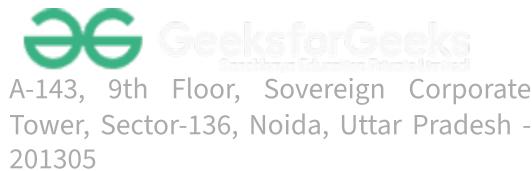
[Top 50 C Coding Interview Questions and Answers \(2024\)](#)

[Share your thoughts in the comments](#)

[Add Your Comment](#)

GeeksforGeeks

Article Tags : [C Language](#)



## Company

[About Us](#)

[Legal](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

[Advertise with us](#)

[GFG Corporate Solution](#)

[Placement Training Program](#)

## Explore

[Job-A-Thon Hiring Challenge](#)

[Hack-A-Thon](#)

[GfG Weekly Contest](#)

[Offline Classes \(Delhi/NCR\)](#)

[DSA in JAVA/C++](#)

[Master System Design](#)

[Master CP](#)

[GeeksforGeeks Videos](#)

[Geeks Community](#)

## Languages

[Python](#)

[Java](#)

[C++](#)

[PHP](#)

[GoLang](#)

[SQL](#)

[R Language](#)

[Android Tutorial](#)

## DSA

[Data Structures](#)

[Algorithms](#)

[DSA for Beginners](#)

[Basic DSA Problems](#)

[DSA Roadmap](#)

[DSA Interview Questions](#)

[Competitive Programming](#)

## Data Science & ML

[Data Science With Python](#)

[Data Science For Beginner](#)

## Web Technologies

[HTML](#)

[CSS](#)

Machine Learning Tutorial	JavaScript
ML Maths	TypeScript
Data Visualisation Tutorial	ReactJS
Pandas Tutorial	NextJS
NumPy Tutorial	NodeJS
NLP Tutorial	Bootstrap
Deep Learning Tutorial	Tailwind CSS

## Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

## Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

## DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

## System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

## School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

## Commerce

Accountancy
Business Studies
Economics
Management
HR Management
Finance
Income Tax

## UPSC Study Material

Polity Notes
Geography Notes
History Notes
Science and Technology Notes
Economy Notes
Ethics Notes
Previous Year Papers

## Preparation Corner

Company-Wise Recruitment Process
Resume Templates
Aptitude Preparation
Puzzles
Company-Wise Preparation
Companies
Colleges

## Competitive Exams

JEE Advanced
UGC NET
SSC CGL

## More Tutorials

Software Development
Software Testing
Product Management

[SBI PO](#)[Project Management](#)[SBI Clerk](#)[Linux](#)[IBPS PO](#)[Excel](#)[IBPS Clerk](#)[All Cheat Sheets](#)

## Free Online Tools

[Typing Test](#)[Image Editor](#)[Code Formatters](#)[Code Converters](#)[Currency Converter](#)[Random Number Generator](#)[Random Password Generator](#)

## Write & Earn

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Share your Experiences](#)[Internships](#)

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

## About Coding Questions

Coding Questions are very important, in both online assessments and technical interviews. Depending on the profile interviewers ask candidates conceptual questions about a program or can ask the candidate to write the whole code for any given question.

## Introduction to Programming

There are primarily four programming languages on which interviewers can ask questions. You can prepare for coding languages here:-

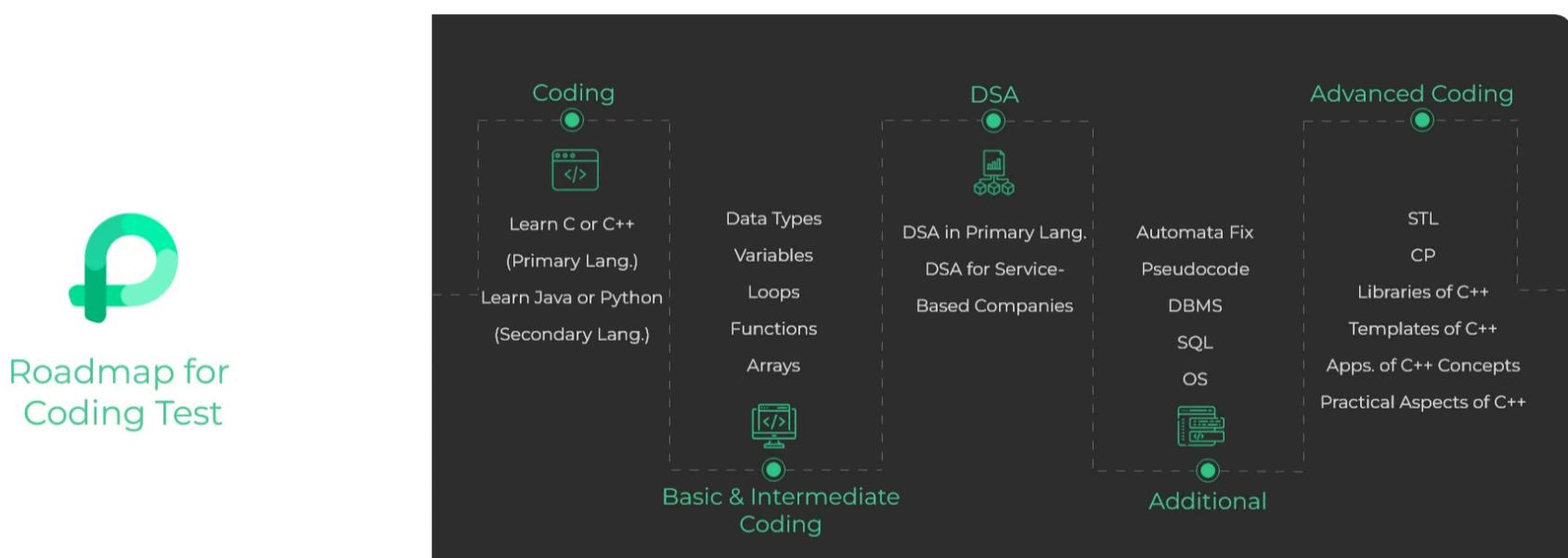
- [Learn C](#)
- [Learn C++](#)
- [Learn Java](#)
- [Learn Python](#)

Other than coding languages, the placement process also requires one to learn about DBMS and DSA.

## Most Asked Coding Questions

Below we have given the most commonly asked coding questions in placement and interviews.

1. [Most asked Coding Questions \(Preplinsta Top 100 Codes\)](#)
2. [Most asked Coding Questions DSA \(Preplinsta Top 100 DSA\)](#)



# Top 50 Coding Interview Questions

### 1. Write a code to reverse a number

C

```
#include<stdio.h>

int main() {
    //Initialization of variables where rev='reverse=0'
    int number, rev = 0, store, left;

    //input a numbers for user
    printf("Enter the number\n");
    scanf("%d", & number);

    store = number;
    //use this loop for check true condition
    while (number > 0) {
        //left is for reminder are left
        left = number % 10;

        //for reverse of no.
        rev = rev * 10 + left;

        //number /= 10;
        number = number / 10;

    }
    //To show the user value
    printf("Given number = %d\n", store);

    //after reverse show numbers
    printf("Its reverse is = %d\n", rev);

    return 0;
}
```

Run

**C++****JAVA****Python**(Use Coupon Code CT10 **and get flat 10% off** on your Preplinsta Prime Subscription.)

Get flat 10% OFF

**2. Write the code to find the Fibonacci series upto the nth term.****C**

```
#include<stdio.h>

int main() {
    int n = 10;
    int a = 0, b = 1;

    // printing the 0th and 1st term
    printf("%d, %d", a, b);

    int nextTerm;

    // printing the rest of the terms here
    for (int i = 2; i < n; i++) {
        nextTerm = a + b;
        a = b;
        b = nextTerm;

        printf("%d, ", nextTerm);
    }

    return 0;
}
```

Run

**C++****Java****Python**

### 3. Write code of Greatest Common Divisor

**C**

```
// The code used a recursive function to return gcd of p and q
int gcd(int p, int q)
{
    // checking divisibility by 0
    if (p == 0)
        return q;

    if (q == 0)
        return p;

    // base case
    if (p == q)
        return p;

    // p is greater
    if (p > q)
        return gcd(p-q, q);

    else
        return gcd(p, q-p);
}

// Driver program to test above function
int main()
{
    int p = 98, q = 56;
    printf("GCD of %d and %d is %d ", p, q, gcd(p, q));
    return 0;
}
```

Run

[C++](#)[JAVA](#)[Python](#)

Use Coupon Code "CT10" and get flat 10% OFF on your Preplinsta Prime subscription.

 Get flat 10% OFF

#### 4. Write code of Perfect number

[C](#)

```
#include<stdio.h>

int main() {
    // Initialization of variables
    int number, i = 1, total = 0;

    // To take user input
    printf("Enter a number: ");
    scanf("%d", & number);

    while (i < number) {
        if (number % i == 0) {
            total = total + i;
            i++;
        }
    }
    //to condition is true
    if (total == number)
        //display
        printf("%d is a perfect number", number);
    //to condition is false
    else
        //display
        printf("%d is not a perfect number", number);

    return 0;
}
```

[Run](#)[C++](#)[JAVA](#)

**Python****5. Write code to Check if two strings are Anagram or not****C**

```
#include<stdio.h>

int main()
{
    //Initializing variables.
    char str[100];
    int i;
    int freq[256] = {0};

    //Accepting inputs.
    printf("Enter the string: ");
    gets(str);

    //Calculating frequency of each character.
    for(i = 0; str[i] != '\0'; i++)
    {
        freq[str[i]]++;
    }

    printf("The non repeating characters are: ");
    for(i = 0; i < 256; i++)
    {
        if(freq[i] == 1)//Finding uniques characters and printing them.
        {
            printf(" %c ", i);
        }
    }
    return 0;
}
```

Run

**C++****JAVA****Python****6. Write code Check if the given string is Palindrome or not****C**

```
#include<stdio.h>
#include <string.h>

int main()
{
    //Initializing variable.
    char str[100];
    int i,length=0,flag=0;

    //Accepting input.
    printf("Enter the string : ");
    gets(str);
    length=strlen(str);

    //Initializing for loop.
    for(i=0;i<length/2;i++)
    {
        //Checking if string is palindrome or not.
        if(str[i]==str[length-i-1])
            flag++;

    }
    //Printing result.
    if(flag==i)
        printf("String entered is palindrome");
    else
        printf("String entered is not palindrome");

    return 0;
}
```

Run

[C++](#)[JAVA](#)[Python](#)

## 7. Write code to Calculate frequency of characters in a string

[C](#)

```
#include<stdio.h>

int main() {
    //Initializing variables.
    char str[100];
    int i;
    int freq[256] = {
        0
    };

    //Accepting inputs.
    printf("Enter the string: ");
    gets(str);

    //Calculating frequency of each character.
    for (i = 0; str[i] != '\0'; i++) {
        freq[str[i]]++;
    }

    //Printing frequency of each character.
    for (i = 0; i < 256; i++) {
        if (freq[i] != 0) {
            printf("The frequency of %c is %d\n", i, freq[i]);
        }
    }
    return 0;
}
```

Run

[C++](#)[JAVA](#)[Python](#)

## 8. Write code to check if two strings match where one string contains wildcard characters

[C](#)

Run

```
#include<stdio.h>
#include<string.h>
bool check(char *str1, char * str2) ;// declaration of the check() function
int main()
{
    char str1[100],str2[100];
    printf("Enter first string with wild characters : ");
    gets(str1);
    printf("Enter second string without wild characters : ");
    gets(str2);
    test(str1,str2);
    return 0;
}

bool check(char *str1, char * str2)
{
    // checking end of both the strings
    if (*str1 == '\0' && *str2 == '\0')
        return true;

    // comparing the characters of both the strings and wild characters(*)
    if (*str1 == '*' && *(str1+1) != '\0' && *str2 == '\0')
        return false;

    // checking wild characters(?)
    if (*str1 == '?' || *str1 == *str2)
        return check(str1+1, str2+1);

    if (*str1 == '*')
        return check(str1+1, str2) || check(str1, str2+1);
    return false;
}

// test() function for running test cases
void test(char *str1, char *str2)
{
    check(str1, str2)? puts(" Yes "): puts(" No ");
}
```

**C++****Python**

## 9. Write a code for bubble sort

**C**

```
#include<stdio.h>

/* Function to print array */
void display(int arr[], int size) {
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

// Main function to run the program
int main() {
    int array[] = {
        5,
        3,
        1,
        9,
        8,
        2,
        4,
        7
    };
    int size = sizeof(array) / sizeof(array[0]);

    printf("Before bubble sort: \n");
    display(array, size);

    int i, j, temp;
    for (i = 0; i < size - 1; i++) {

        // Since, after each iteration rightmost i elements are sorted
        for (j = 0; j < size - i - 1; j++)
            if (array[j] > array[j + 1]) {
                temp = array[j]; // swap the element
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
    }
    printf("After bubble sort: \n");
    display(array, size);
    return 0;
}
```

Run

## C++

### 10. How is the merge sort algorithm implemented?

C

Run

```
#include<stdio.h>

void mergeSort(int[], int, int);
void merge(int[], int, int, int);

void display(int arr[], int size) {
    int i;
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void main() {
    int a[10] = {
        11,
        9,
        6,
        19,
        33,
        64,
        15,
        75,
        67,
        88
    };
    int i;

    int size = sizeof(a) / sizeof(a[0]);
    display(a, size);

    mergeSort(a, 0, size - 1);
    display(a, size);
}

void mergeSort(int a[], int left, int right) {
    int mid;
    if (left < right) {
        // can also use mid = left + (right - left) / 2
        // this can avoid data type overflow
        mid = (left + right) / 2;

        // recursive calls to sort first half and second half subarrays
        mergeSort(a, left, mid);
        mergeSort(a, mid + 1, right);
        merge(a, left, mid, right);
    }
}

void merge(int arr[], int left, int mid, int right) {
    int i, j, k;
    int n1 = mid - left + 1;
    int n2 = right - mid;

    // create temp arrays to store left and right subarrays
    int L[n1], R[n2];

    // Copying data to temp arrays L[] and R[]
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    // here we merge the temp arrays back into arr[l..r]
    i = 0; // Starting index of L[i]
    j = 0; // Starting index of R[i]
    k = left; // Starting index of merged subarray

    while (i < n1 && j < n2) {
        // place the smaller item at arr[k] pos
        if (L[i] < R[j])
            arr[k] = L[i];
        else
            arr[k] = R[j];
        i++;
        j++;
        k++;
    }

    // If there are any remaining elements in L[]
    while (i < n1)
        arr[k] = L[i];
        i++;
        k++;

    // If there are any remaining elements in R[]
    while (j < n2)
        arr[k] = R[j];
        j++;
        k++;
}
}
```

```

if (L[i] <= R[j]) {
    arr[k] = L[i];
    i++;
} else {
    arr[k] = R[j];
    j++;
}
k++;

}

// Copy the remaining elements of L[], if any
while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
}

// Copy the remaining elements of R[], if any
while (j < n2) {
    arr[k] = R[j];
    j++;
    k++;
}
}

```

**C++****JAVA**

### 11. Write to code to check whether a given year is leap year or not.

**C**

```

#include <stdio.h>
Run

int main() {
    int year;
    scanf("%d", & year);

    if (year % 400 == 0)
        printf("%d is a Leap Year", year);

    else if (year % 4 == 0 && year % 100 != 0)
        printf("%d is a Leap Year", year);

    else
        printf("%d is not a Leap Year", year);

    return 0;
}

```

**C++**

## 12. Find non-repeating characters in a string

C

```
#include<stdio.h>
int main()
{
//Initializing variables.
char str[100]={"prepinsta"};
int i;
int freq[256] = {0};
//Calculating frequency of each character.
for(i = 0; str[i] != '\0'; i++)
{
freq[str[i]]++;
}
printf("The non repeating characters are: ");
for(i = 0; i < 256; i++)
{
if(freq[i] == 1)//Finding uniques characters and printing them.
{
printf(" %c ", i);
}
}
return 0;
}
```

C++

JAVA

Python

## 13. Write a code to replace a substring in a string.

C

```

#include<stdio.h>
#include<string.h>
int main() {
    char str[256] = "preinsta", substr[128] = "insta", replace[128] = "ster ", output[256];
    int i = 0, j = 0, flag = 0, start = 0;

    str[strlen(str) - 1] = '\0';
    substr[strlen(substr) - 1] = '\0';
    replace[strlen(replace) - 1] = '\0';

    // check whether the substring to be replaced is present
    while (str[i] != '\0')
    {
        if (str[i] == substr[j])
        {
            if (!flag)
                start = i;
            j++;
            if (substr[j] == '\0')
                break;
            flag = 1;
        }
        else
        {
            flag = start = j = 0;
        }
        i++;
    }
    if (substr[j] == '\0' && flag)
    {
        for (i = 0; i < start; i++)
            output[i] = str[i];

        // replace substring with another string
        for (j = 0; j < strlen(replace); j++)
        {
            output[i] = replace[j];
            i++;
        }
        // copy remaining portion of the input string "str"
        for (j = start + strlen(substr); j < strlen(str); j++)
        {
            output[i] = str[j];
            i++;
        }
        // print the final string
        output[i] = '\0';
        printf("Output: %s\n", output);
    } else {
        printf("%s is not a substring of %s\n", substr, str);
    }
    return 0;
}

```

**C++****JAVA****Python**

Use Coupon Code “**CT10**” and get flat 10% OFF on your Preplinsta Prime subscription.

 Get flat 10% OFF

## 14. Write a code for Heap sort.

C

```
#include<stdio.h> // including Library files
int temp;

void heapify(int arr[], int size, int i)//declaring functions
{
    int max = i;
    int left = 2*i + 1;
    int right = 2*i + 2;

    if (left < size && arr[left] >arr[max])
        max= left;

    if (right < size && arr[right] > arr[max])
        max= right;

    if (max!= i)
    {
        // performing sorting Logic by using temporary variable
        temp = arr[i];
        arr[i]= arr[max];
        arr[max] = temp;
        heapify(arr, size, max);
    }
}

void heapSort(int arr[], int size)// providing definition to heap sort
{
    int i;
    for (i = size / 2 - 1; i >= 0; i--)
        heapify(arr, size, i);
    for (i=size-1; i>=0; i--)
    {
        // swaping logic
        temp = arr[0];
        arr[0]= arr[i];
        arr[i] = temp;
        heapify(arr, i, 0);
    }
}

void main() // defining main()
{
    int arr[] = {58, 134, 3, 67, 32, 89, 15, 10, 78, 9};
    // array initializing with their elements.
    int i;
    int size = sizeof(arr)/sizeof(arr[0]);

    heapSort(arr, size);

    printf("printing sorted elements\n"); // printing the sorted array
    for (i=0; i<size; ++i)
        printf("%d\n",arr[i]);
}
```

C++

JAVA

## 15. Write a code to replace each element in an array by its rank in the array

C

```
#include<stdio.h>

int main(){
    int arr[] = { 100, 2, 70, 12 , 90};
    int n = sizeof(arr) / sizeof(arr[0]);

    int temp[n];
    for(int i=0; i<n; i++)
        temp[i] = arr[i];

    //sort the copied array
    for(int i=0; i<n; i++){
        for(int j=i+1; j<n; j++){
            int x = temp[i];
            temp[i] = temp[j];
            temp[j] = x;
        }
    }

    for(int i=0; i<n; i++){

        for(int j=0; j<n; j++){
            if(temp[j]==arr[i])
            {
                arr[i] = j+1;
                break;
            }
        }
    }

    for(int i=0; i<n; i++)
        printf("%d ", arr[i]);
}
```

C++

JAVA

Python

#### 16. Write a code to find circular rotation of an array by K positions.

C

```
#include<stdio.h>
int main()
{
    int size;
    printf("Size of array: ");
    scanf("%d",&size);
    int arr[size];
    printf("Enter the elements ");
    for(int a=0;a<size;a++)
        scanf("%d",&arr[a]);
    int n;
    printf("Enter the index from where you want your array to rotate ");
    scanf("%d",&n);
    printf("Array: \n");
    for (int a = 0; a < size; a++) {
        printf("%d ", arr[a]);
    }
    for(int a = 0; a < n; a++) { int b, temporary; temporary = arr[size-1]; for(b = size-1; b > 0; b--)
    {
        arr[b] = arr[b-1];
    }
    arr[0] = temporary;
}
    printf("\n");
    printf("New Array: \n");
    for(int a = 0; a< size;a++){
        printf("%d ", arr[a]);
    }
    return 0;
}
```

C++

JAVA

Python

## 17. Write a code to find non-repeating elements in an array.

C

```
#include<stdio.h>

// Main function to run the program
int main()
{
    int arr[] = {21, 30, 10, 2, 10, 20, 30, 11};
    int n = sizeof(arr)/sizeof(arr[0]);

    int visited[n];

    for(int i=0; i<n; i++){

        if(visited[i]==0){
            int count = 1;
            for(int j=i+1; j<n; j++){
                if(arr[i]==arr[j]){
                    count++;
                    visited[j]=1;
                }
            }
            if(count==1)
                printf("%d %arr[i]);
        }
    }

    return 0;
}
```

C++

JAVA

Python

## 18. Write a code to check for the longest palindrome in an array.

C

```
#include<stdio.h>
#include<limits.h>
int ispalindrome(int n){
    int rev=0, temp = n;

    while(temp>0){
        int rem = temp%10;
        rev = rev*10 + rem;
        temp /= 10;
    }

    if(n==rev)
        return 1;

    return 0;
}

int main(){
    int arr[] = {1, 121, 5551, 545545, 10111, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    int res = INT_MIN;

    for(int i=0; i<n; i++){
        if(ispalindrome(arr[i]) && res<arr[i])
            res = arr[i];
    }

    if(res==INT_MIN)
        res = -1;

    printf("%d ",res);
}
```

**C++****JAVA****Python****19. Write a code to find the factorial of a number.****C**

```
#include<stdio.h>
int main ()
{
    int num = 5, fact = 1;

    // Can't calculate factorial of a negative number
    if(num < 0)
        printf("Error");
    else
    {
        for(int i = 1; i <= num; i++)
            fact = fact * i;
    }

    printf("Fact %d: %d",num, fact);
}
// Time complexity: O(N)
// Space complexity: O(1)
```

[C++](#)[JAVA](#)[Python](#)**20. Write the code to for Armstrong number**[C](#)

```

#include
#include

// Armstrong number is any number following the given rule
// abcd... = a^n + b^n + c^n + d^n + ...
// Where n is the order(length/digits in number)

// Example = 153 (order/length = 3)
// 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153

// Example = 1634 (order/length = 4)
// 1634 = 1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634

// number of digits in a number is order
int order(int x)
{
    int len = 0;
    while (x)
    {
        len++;
        x = x/10;
    }
    return len;
}

int armstrong(int num, int len){

    int sum = 0, temp, digit;
    temp = num;

    // loop to extract digit, find power & add to sum
    while(temp != 0)
    {
        // extract digit
        digit = temp % 10;

        // add power to sum
        sum = sum + pow(digit,len);
        temp /= 10;
    };

    return num == sum;
}

// Driver Code
int main ()
{
    int num, len;

    printf("Enter a number: ");
    scanf("%d",&num);

    // function to get order(length)
    len = order(num);

    // check if Armstrong
    if (armstrong(num, len))
        printf("%d is Armstrong", num);
    else
        printf("%d is Not Armstrong", num);

}

```

**C++****JAVA**

**Python****21. Write a program to find the sum of Natural Numbers using Recursion.****C**

```
#include<stdio.h>

int getSum(int sum,int n)
{
    if(n==0)
        return sum;

    return n+getSum(sum,n-1);
}

int main()
{
    int n, sum = 0;
    scanf("%d",&n);

    printf("%d",getSum(sum, n));

    return 0;
}
// Time complexity : O(n)
// Space complexity : O(1)
// Auxilaray space complexity : O(N)
// Due to function call stack
```

**C++****JAVA****Python****22. Write a program to add Two Matrices using Multi-dimensional Array.****C**

```
#include
int main() {
int r, c, a[100][100], b[100][100], sum[100][100], i, j;
printf("Enter the number of rows (between 1 and 100): ");
scanf("%d", &r);
printf("Enter the number of columns (between 1 and 100): ");
scanf("%d", &c);

printf("\nEnter elements of 1st matrix:\n");
for (i = 0; i < r; ++i)
for (j = 0; j < c; ++j) {
printf("Enter element a%d%d: ", i + 1, j + 1);
scanf("%d", &a[i][j]);
}

printf("\nEnter elements of 2nd matrix:\n");
for (i = 0; i < r; ++i)
for (j = 0; j < c; ++j) {
printf("Enter element a%d%d: ", i + 1, j + 1);
scanf("%d", &b[i][j]);
}

// adding two matrices
for (i = 0; i < r; ++i)
for (j = 0; j < c; ++j) {
sum[i][j] = a[i][j] + b[i][j];
}

// printing the result
printf("\nSum of two matrices: \n");
for (i = 0; i < r; ++i)
for (j = 0; j < c; ++j) {
printf("%d ", sum[i][j]);
if (j == c - 1) {
printf("\n\n");
}
}
return 0;
}
```

### 23. Write a Program to Find the Sum of Natural Numbers using Recursion.

C

```
#include
Numbers(int n);

int main() {

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

printf("Sum = %d", addNumbers(num));

return 0;
}

int addNumbers(int n) {

if (n != 0)

return n + addNumbers(n - 1);

else

return n;
}
```

[C++](#)[JAVA](#)[Python](#)**24. Write code to check a String is palindrome or not?**

C

```
#include
#include

// A function to check if a string str is palindrome
void isPalindrome(char str[])
{
    // Start from leftmost and rightmost corners of str
    int l = 0;
    int h = strlen(str) - 1;

    // Keep comparing characters while they are same
    while (h > l)
    {
        if (str[l++] != str[h--])
        {
            printf("%s is Not Palindrome", str);
            return;
        }
    }
    printf("%s is palindrome", str);
}

// Driver program to test above function
int main()
{
    isPalindrome("abba");
    isPalindrome("abbccbba");
    isPalindrome("geeks");
    return 0;
}
```

**C++****JAVA****Python**

## 25. Write a program for Binary to Decimal conversion

**C**

```
#include<stdio.h>

int main()
{
    int num, binary_val, decimal_val = 0, base = 1, rem;

    printf("Insert a binary num (1s and 0s) \n");
    scanf("%d", &num); /* maximum five digits */

    binary_val = num;
    while (num > 0)
    {
        rem = num % 10;
        decimal_val = decimal_val + rem * base;
        //num/=10;
        num = num / 10;
        //base*=2;
        base = base * 2;
    }
    //display binary number
    printf("The Binary num is = %d \n", binary_val);
    //display decimal number
    printf("Its decimal equivalent is = %d \n", decimal_val);
    return 0;
}
```

**C++****JAVA****Python****26. Write a program to check whether a character is a vowel or consonant****C**

```
#include
int main()
{
char c;
int isLowerVowel, isUpperVowel;
printf("Enter an alphabet: ");
scanf("%c",&c);

//To find the corrector is lowercase vowel
isLowerVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
//To find the character is Upper case vowel
isUpperVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
// compare to charector is Lowercase Vowel or Upper case Vowel

if (isLowerVowel || isUpperVowel)
printf("%c is a vowel", c);
//to check character is alphabet or not

elseif((c >= 'a' && c= 'A' && c <= 'Z'))
printf("\n not a alphabet\n");

else
printf("%c is a consonant", c);

return 0;
}
```

**C++****JAVA****Python****27. Write a code to find an Automorphic number****C**

```
#include<stdio.h>

int checkAutomorphic(int num)
{
    int square = num * num;

    while (num > 0)
    {
        if (num % 10 != square % 10)
            return 0;

        // Reduce N and square
        num = num / 10;
        square = square / 10;
    }
    return 1;
}

int main()
{
    //enter value
    int num;
    scanf("%d",&num);

    //checking condition
    if(checkAutomorphic(num))
        printf("Automorphic");
    else
        printf("Not Automorphic");
    return 0;
}
```

**C++****JAVA****28. Write a code to find Find the ASCII value of a character****C**

```
/* C Program to identify ASCII Value of a Character */
#include
#include
int main()
{
    char a;

    printf("\n Kindly insert any character \n");
    scanf("%c",&a);

    printf("\n The ASCII value of inserted character = %d",a);
    return 0;
}
```

**C++****JAVA****Python****29. Write a code to Remove all characters from string except alphabets****C**

```
#include <stdio.h>
int main()
{
    //Initializing variable.
    char str[100];
    int i, j;

    //Accepting input.
    printf(" Enter a string : ");
    gets(str);

    //Iterating each character and removing non alphabetical characters.
    for(i = 0; str[i] != '\0'; ++i)
    {
        while (!( (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || str[i] == '\0' ) )
        {
            for(j = i; str[j] != '\0'; ++j)
            {
                str[j] = str[j+1];
            }
            str[j] = '\0';
        }
        //Printing output.
        printf(" After removing non alphabetical characters the string is :");
        puts(str);
    }
    return 0;
}
```

C++

JAVA

Python

Use Coupon Code "CT10" and get flat 10% OFF on your Preplinsta Prime subscription.

 Get flat 10% OFF

### 30. Write a code to Print the smallest element of the array

C

```
#include < stdio.h >

int getSmallest(int arr[], int len)
{
    // assign first array element as smallest
    int min = arr[0];

    // linearly search for the smallest element
    for(int i=1; i < len; i++)
    {
        // if the current array element is smaller
        if (arr[i] < min)
            min = arr[i];
    }

    return min;
}

int main()
{
    int arr[] = {5, 8, 7, 2, 12, 4};

    // get the length of the array
    int len = sizeof(arr)/sizeof(arr[0]);

    printf("The smallest : %d", getSmallest(arr, len));
}
```

**C++****JAVA****Python****31. Write a code to Reverse the element of the array****C**

```
#include < stdio.h>

void printReverse(int arr[], int len){

    for(int i = len - 1; i >= 0; i--)
        printf("%d ", arr[i]);
}

int main()
{
    int arr[] = {10, 20, 30, 40, 50, 60};

    int len = sizeof(arr)/sizeof(arr[0]);

    printf("Array in Reverse:\n");
    printReverse(arr, len);

    return 0;
}
```

**C++****JAVA****Python****32. Write a code to Sort the element of the array**

**C**

```
#include < stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int array[], int size)
{
    int i, j, min_idx;

    // Loop to iterate on array
    for (i = 0; i < size-1; i++)
    {
        // Here we try to find the min element in array
        min_idx = i;
        for (j = i+1; j < size; j++)
        {
            if (array[j] < array[min_idx])
                min_idx = j;
        }
        // Here we interchange the min element with first one
        swap(&array[min_idx], &array[i]);
    }
}

/* Display function to print values */
void display(int array[], int size)
{
    int i;
    for (i=0; i < size; i++)
    {
        printf("%d ",array[i]);
    }
    printf("\n");
}

// The main function to drive other functions
int main()
{
    int array[] = {50, 30, 10, 90, 80, 20, 40, 70};
    int size = sizeof(array)/sizeof(array[0]);

    selectionSort(array, size);

    display(array, size);

    return 0;
}
```

**C++****JAVA****Python****33. Write a code to Sort the element of the array without sort method****C**

```
#include < stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int array[], int size)
{
    int i, j, min_idx;

    // Loop to iterate on array
    for (i = 0; i < size-1; i++)
    {
        // Here we try to find the min element in array
        min_idx = i;
        for (j = i+1; j < size; j++)
        {
            if (array[j] < array[min_idx])
                min_idx = j;
        }
        // Here we interchange the min element with first one
        swap(&array[min_idx], &array[i]);
    }
}

/* Display function to print values */
void display(int array[], int size)
{
    int i;
    for (i=0; i < size; i++)
    {
        printf("%d ",array[i]);
    }
    printf("\n");
}

// The main function to drive other functions
int main()
{
    int array[] = {50, 30, 10, 90, 80, 20, 40, 70};
    int size = sizeof(array)/sizeof(array[0]);

    selectionSort(array, size);

    display(array, size);

    return 0;
}
```

**C++****JAVA****Python****34. Write a code to Replace a Substring in a string****C**

```
#include< stdio.h>
#include< string.h>
int main() {
    char str[256] = "prepinsta", substr[128] = "insta", replace[128] = "ster ", output[256];
    int i = 0, j = 0, flag = 0, start = 0;

    str[strlen(str) - 1] = '\0';
    substr[strlen(substr) - 1] = '\0';
    replace[strlen(replace) - 1] = '\0';

    // check whether the substring to be replaced is present
    while (str[i] != '\0')
    {
        if (str[i] == substr[j])
        {
            if (!flag)
                start = i;
            j++;
            if (substr[j] == '\0')
                break;
            flag = 1;
        }
        else
        {
            flag = start = j = 0;
        }
        i++;
    }
    if (substr[j] == '\0' && flag)
    {
        for (i = 0; i < start; i++)
            output[i] = str[i];

        // replace substring with another string
        for (j = 0; j < strlen(replace); j++)
        {
            output[i] = replace[j];
            i++;
        }
        // copy remaining portion of the input string "str"
        for (j = start + strlen(substr); j < strlen(str); j++)
        {
            output[i] = str[j];
            i++;
        }
        // print the final string
        output[i] = '\0';
        printf("Output: %s\n", output);
    } else {
        printf("%s is not a substring of %s\n", substr, str);
    }
    return 0;
}
```

**C++****JAVA****Python****35. Write a code to Remove space from a string****C**

```
#include< stdio.h>
using namespace std;
// Function to remove all spaces from a given string
void removeSpaces(char *str)
{
    // To keep track of non-space character count
    int count = 0;
    // Traverse the provided string. If the current character is not a space,
    // move it to index 'count++'.
    for (int i = 0; str[i]; i++)
        if (str[i] != ' ')
            str[count++] = str[i]; // here count is incremented
    str[count] = '\0';
}
// Driver program to test above function
int main()
{
    char str[] = "P re p i n sta ";
    removeSpaces(str);
    printf("%s", str);
    return 0;
}
```

**C++**

**JAVA**

**Python**

### 36. Write a code to Count Inversion

**C**

```
#include < stdio.h >

int _mergeSort(int arr[], int temp[], int left, int right);
int merge(int arr[], int temp[], int left, int mid,int right);

/* This function sorts the
input array and returns the
number of inversions in the array */
int mergeSort(int arr[], int array_size)
{
    int temp[array_size];
    return _mergeSort(arr, temp, 0, array_size - 1);
}

/* An auxiliary recursive function
that sorts the input array and
returns the number of inversions in the array. */
int _mergeSort(int arr[], int temp[], int left, int right)
{
    int mid, inv_count = 0;
    if (right > left) {
        /* Divide the array into two parts and
        call _mergeSortAndCountInv()
        for each of the parts */
        mid = (right + left) / 2;

        /* Inversion count will be sum of
        inversions in left-part, right-part
        and number of inversions in merging */
        inv_count += _mergeSort(arr, temp, left, mid);
        inv_count += _mergeSort(arr, temp, mid + 1, right);

        /*Merge the two parts*/
        inv_count += merge(arr, temp, left, mid + 1, right);
    }
    return inv_count;
}

/* This funt merges two sorted arrays
and returns inversion count in the arrays.*/
int merge(int arr[], int temp[], int left, int mid,int right)
{
    int i, j, k;
    int inv_count = 0;

    i = left; /* i is index for left subarray*/
    j = mid; /* j is index for right subarray*/
    k = left; /* k is index for resultant merged subarray*/
    while ((i <= mid - 1) && (j <= right)) {
        if (arr[i] <= arr[j]) {
            temp[k++] = arr[i++];
        }
        else {
            temp[k++] = arr[j++];

            /* this is tricky -- see above
            explanation/diagram for merge()*/
            inv_count = inv_count + (mid - i);
        }
    }

    /* Copy the remaining elements of left subarray
    (if there are any) to temp*/
    while (i <= mid - 1)
        temp[k++] = arr[i++];

    /* Copy the remaining elements of right subarray
    (if there are any) to temp*/
    while (j <= right)
        temp[k++] = arr[j++];

    /*Copy back the merged elements to original array*/
    for (i = left; i <= right; i++)

```

```

        arr[i] = temp[i];

        return inv_count;
    }

// Driver code
int main()
{
    int n ;

    scanf("%d", &n);

    int arr[n];
    for(int i=0; i

```

**C++****JAVA****Python**

### 37. Write a code to find consecutive largest subsequence

**C**

```

#include < stdio.h>
#include < stdlib.h>

//call back function
int compare(const void * a, const void * b)
{
    return ( *(int*)a - *(int*)b );
}

int findLongestConseqSeq(int arr[], const int n)
{
    int length = 1;
    int longestConsecutiveSeq = 1;
    int i =0;

    //sort arr elements using qsort inbuilt function
    qsort( arr,n, sizeof(int), compare);
    for ( i = 0; i < n - 1; i++) {
        if(arr[i] == arr[i+1]) {
            continue;
        }
        else if (arr[i] + 1 == arr[i + 1]) {
            length++;
        }
        else {
            length = 1;
        }
        longestConsecutiveSeq = (longestConsecutiveSeq > length)? longestConsecutiveSeq: length;
    }
    return longestConsecutiveSeq;
}

int main()
{
    int arr[] = {2,5,7,7,8,8,9,4,10,12,3,6};

    const int N = sizeof(arr)/sizeof(arr[0]);
    const int longestConsecutiveSeq = findLongestConseqSeq(arr, N);

    printf("Longest Consecutive Sequence is %d",longestConsecutiveSeq);
    return 0;
}

```

[C++](#)[JAVA](#)[Python](#)

### 38: Write a Program to Find out the Sum of Digits of a Number.

**Program in C**

```
#include

int main ()
{
    int num, sum = 0;

    printf("Enter any num: ");
    scanf("%d",&num);

    //loop to find sum of digits
    while(num!=0){
        sum += num % 10;
        num = num / 10;
    }

    //output
    printf("Sum: %d",sum);

    return 0;
}

// Time complexity : O(N)
// Space complexity : O(1)
```

Find More Solutions at [C Program to Find the Sum of Digits of a Number](#)

**Program in C++****Program in JAVA****Program in Python**

### 39: Write a Program to Find out the Power of a Number

**Program in C**

```
// pow function is contained in math.h library
#include<stdio.h>
#include <math.h>

int main()
{
    double base = 2.3;
    double exp = 2.1;
    double result;

    // calculates the power
    result = pow(base, exp);

    // %lf used for double
    printf("%lf ^ %lf = %lf\n", base, exp, result);

    // following can be used for precision setting
    printf("%.1lf ^ %.1lf = %.2lf", base, exp, result);

    return 0;
}
```

Find More Solutions at [C Program to find out the Power of a Number](#)

#### **Program in C++**

#### **Program in JAVA**

#### **Program in Python**

### **40: Write a Program to Find out the Sum of Digits of a Number.**

#### **Program in C**

```
#include

int main ()
{
    int num, sum = 0;

    printf("Enter any num: ");
    scanf("%d",&num);

    //loop to find sum of digits
    while(num!=0){
        sum += num % 10;
        num = num / 10;
    }

    //output
    printf("Sum: %d",sum);

    return 0;
}

// Time complexity : O(N)
// Space complexity : O(1)
```

Find More Solutions at [C Program to Find the Sum of Digits of a Number](#)

#### **Program in C++**

#### **Program in JAVA**

**Program in Python****41: Write a Program to Add two Fractions****Program in C**

```
#include<stdio.h>
int main()
{
//for initialize variables
int numerator1, denominator1, numerator2, denominator2, x, y, c, gcd_no;

//To take user input of numerators and denominators
printf("Enter the numerator for 1st number : ");
scanf("%d",&numerator1);
printf("Enter the denominator for 1st number : ");
scanf("%d",&denominator1);
printf("Enter the numerator for 2nd number : ");
scanf("%d",&numerator2);
printf("Enter the denominator for 2nd number : ");
scanf("%d",&denominator2);

//numerator
x=(numerator1*denominator2)+(denominator1*numerator2);

//denominator
y=denominator1*denominator2;

// Trick part. Reduce it to the simplest form by using gcd.
for(c=1; c <= x && c <= y; ++c)
{
if(x%c==0 && y%c==0)
gcd_no = c;
}

//To display fraction of givien numerators and denominators
printf("(%d / %d) + (%d / %d) = (%d / %d)", numerator1, denominator1, numerator2, denominator2, x/gcd_no,
y/gcd_no);

return 0;
}
```

Find More Solutions at [C Program to add two fractions](#)

**Program in C++****Program in JAVA****Program in Python****42: Write a Program to Find the Largest Element in an Array.****Program in C**

```
// C Program to find largest element in an array
#include<stdio.h>

int getLargest(int arr[], int len)
{
    // assign first array element as largest
    int max = arr[0];

    // linearly search for the largest element
    for(int i=1; i < len; i++)
        max = arr[i];
}

return max;
}

int main()
{
    int arr[] = {20, 5, 35, 40, 10, 50, 15};

    // get the length of the array
    int len = sizeof(arr)/sizeof(arr[0]);

    printf("The Largest element is: %d", getLargest(arr, len));
}

// Time complexity: O(N)
// Space complexity: O(1)
```

Find More Solutions at [C Program to Find the Largest Element in an Array](#)

**Program in C++**

**Program in JAVA**

**Program in Python**

#### 43: Write a Program to Find the Roots of a Quadratic Equation

**Program in C**

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

void findRoots(int a, int b, int c)
{
    if (a == 0) {
        printf("Invalid");
        return;
    }

    int d = b * b - 4 * a * c;
    double sqrt_val = sqrt(abs(d));

    if (d > 0) {
        printf("Roots are real and different \n");
        printf("%f\n%f", (double)(-b + sqrt_val) / (2 * a), (double)(-b - sqrt_val) / (2 * a));
    }
    else if (d == 0) {
        printf("Roots are real and same \n");
        printf("%f", -(double)b / (2 * a));
    }
    else // d < 0
    {
        printf("Roots are complex \n");
        printf("%f + %fi\n%f - %fi", -(double)b / (2 * a), sqrt_val/(2 * a), -(double)b / (2 * a), sqrt_val/(2 * a));
    }
}

int main()

{
    int a = 1, b = 4, c = 4;

    findRoots(a, b, c);
    return 0;
}
```

Find More Solutions at [C Program to Find the Roots of a Quadratic Equation](#)

**Program in C++**

**Program in JAVA**

**Program in Python**

#### 44: Write a Program to Find the Prime Factors of a Number.

**Program in C**

```
#include<stdio.h>

void primefactor(int num) {

    printf("Prime factors of the number : ");
    for (int i = 2; num > 1; i++) {

        while (num % i == 0) {
            printf("%d ", i);
            num = num / i;
        }
    }
}

int main() {

    int num;
    printf("Enter the positive integer: ");
    scanf("%d", &num);

    primefactor(num);
    return 0;
}
```

Find More Solutions at [C Program to Find the Prime Factors of a Number](#)

**Program in C++**

**Program in JAVA**

**Program in Python**

#### 45: Write a Program to Convert Digits to Words.

**Program in C**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void convert_to_words(char* num)
{
    int len = strlen(num);

    /* Base cases */
    if (len == 0) {
        fprintf(stderr, "empty string\n");
        return;
    }
    if (len > 4) {
        fprintf(stderr,
        "Length more than 4 is not supported\n");
        return;
    }

    char* single_digits[] = { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" };
    char* two_digits[] = { "", "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen",
    "seventeen", "eighteen", "nineteen" };

    char* tens_multiple[] = { "", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty",
    "ninety" };

    char* tens_power[] = { "hundred", "thousand" };

    printf("\n%s: ", num);

    if (len == 1) {
        printf("%s\n", single_digits[*num - '0']);
        return;
    }

    while (*num != '\0') {

        if (len >= 3) {
            if (*num - '0' != 0) {
                printf("%s ", single_digits[*num - '0']);
                printf("%s ", tens_power[len - 3]);
            }
            --len;
        }

        else {
            if (*num == '1') {
                int sum = *num - '0' + *(num + 1) - '0';
                printf("%s\n", two_digits[sum]);
                return;
            }

            else if (*num == '2' && *(num + 1) == '0') {
                printf("twenty\n");
                return;
            }

            else {
                int i = *num - '0';
                printf("%s ", i ? tens_multiple[i] : "");
                ++num;
                if (*num != '0')
                    printf("%s ",
                    single_digits[*num - '0']));
            }
        }
        ++num;
    }
}
```

```
}

int main(void)
{
    convert_to_words("9459");
    return 0;
}
```

Find More Solutions at [C Program to Convert Digits to Words](#)

**Program in C++**

**Program in JAVA**

**Program in Python**

#### 46: Write a Program to Find the Factorial of a Number using Recursion.

**Program in C**

```
#include<stdio.h>
int getFactorial(int num)
{
    if(num == 0)
        return 1;

    return num * getFactorial(num-1);
}

int main ()
{
    int num = 7;

    int fact = getFactorial(num);

    printf("Fact %d: %d",num, fact);
}
```

Find More Solutions at [C Program to Find the Factorial of a Number using Recursion](#)

**Program in C++**

**Program in JAVA**

**Program in Python**

#### 47: Write a Program to Reverse an Array

**Program in C**

```
#include <stdio.h>

void printReverse(int arr[], int len){

    for(int i = len - 1; i >= 0; i--)
        printf("%d ", arr[i]);
}

int main()
{
    int arr[] = {10, 20, 30, 40, 50, 60};

    int len = sizeof(arr)/sizeof(arr[0]);

    printf("Array in Reverse:\n");
    printReverse(arr, len);

    return 0;
}
```

Find More Solutions at [C Program to Reverse an Array](#)

**Program in C++**

**Program in JAVA**

**Program in Python**

#### 48. Write code to check if two strings match where one string contains wildcard characters

C

```

#include
#include
bool check(char *str1, char * str2) ;// declaration of the check() function
int main()
{
    char str1[100],str2[100];
    printf("Enter first string with wild characters : ");
    gets(str1);
    printf("Enter second string without wild characters : ");
    gets(str2);
    test(str1,str2);
    return 0;
}

bool check(char *str1, char * str2)
{
    // checking end of both the strings
    if (*str1 == '\0' && *str2 == '\0')
        return true;

    // comparing the characters of both the strings and wild characters(*)
    if (*str1 == '*' && *(str1+1) != '\0' && *str2 == '\0')
        return false;

    // checking wild characters(?)
    if (*str1 == '?')
        return check(str1+1, str2) || check(str1, str2+1);

    if (*str1 == '*')
        return check(str1+1, str2) || check(str1, str2+1);
    return false;
}

// test() function for running test cases
void test(char *str1, char *str2)
{
    check(str1, str2)? puts(" Yes "): puts(" No ");
}

```

**C++****Python****49: Write a Program to find out the Spiral Traversal of a Matrix.****Program in C**

```

#include <stdio.h>
#define r 4
#define c 4

int main()
{
    int a[4][4] = { { 1, 2, 3, 4 },
                    { 5, 6, 7, 8 },
                    { 9, 10, 11, 12 },
                    { 13, 14, 15, 16 } };

    int i, left = 0, right = c-1, top = 0, bottom = r-1;

    while (left <= right && top <= bottom) {

        /* Print the first row
        from the remaining rows */
        for (i = left; i <= right; ++i) {
            printf("%d ", a[top][i]);
        }
        top++;

        /* Print the last column
        from the remaining columns */
        for (i = top; i <= bottom; ++i) {
            printf("%d ", a[i][right]);
        }
        right--;

        /* Print the last row from
        the remaining rows */
        if (top <= bottom) { for (i = right; i >= left; --i) {
            printf("%d ", a[bottom][i]);
        }
        bottom--;
    }

        /* Print the first column from
        the remaining columns */
        if (left <= right) { for (i = bottom; i >= top; --i) {
            printf("%d ", a[i][left]);
        }
        left++;
    }

    return 0;
}

```

Find More Solutions at [C Program to Find the Spiral Traversal of a Matrix](#)

**Program in C++**

**Program in JAVA**

## 50. Write a code to find Fibonacci Series using Recursion

**C**

```
//Fibonacci Series using Recursion
#include
int fibo(int n)
{
    if (n <= 1)
        return n;
    return fibo(n-1) + fibo(n-2);
}
```

```
int main ()
{
    int n = 9;
    printf("%d", fib(n));
    getchar();
    return 0;
}
```

C++

JAVA

Python

## FAQs on Most Asked Coding Questions in Placement

### Question 1: What are the basics of coding?

The basics of coding are Data Types, Flow controls (loops) and functional programming.

### Question 2: Which company has the hardest coding interview questions?

Product Based Companies like Google, Microsoft, Amazon ask the hardest coding interview questions.

### Question 3: What language is used in coding?

Coding languages mostly used in placement exams are Java, C, C++ and Python.

# Get over 200+ course One Subscription

Courses like AI/ML, Cloud Computing, Ethical Hacking, C, C++, Java, Python, DSA (All Languages), Competitive Coding (All Languages), TCS, Infosys, Wipro, Amazon, DBMS, SQL and others

Get Prime

Course List



The coding interview questions addresses in this article are grouped into 2 categories (as below) to make your learning simpler.

1. Conceptual Interview Questions
2. Programming Interview Questions

We will now look at the first category of coding interview questions.

## Coding Interview Questions On Conceptual Understanding

This section covers some coding interview questions that test the conceptual understanding of the candidate.

### 1. What is a Data Structure?

- A **data structure** is a storage format that defines the way data is stored, organized, and manipulated.
- Some popular data structures are Arrays, Trees, and Graphs.

## Become a Software Development Professional



**Full Stack Java Developer**



**Full Stack Web Developer - MEAN Stack**

Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts

Hands-on practice through 20+ projects, assessments, and tests

6 months

[View Program](#)

Comprehensive Blended Learning program

8X higher interaction in live online classes conducted by industry experts

11 months

[View Program](#)

Not sure what you're looking for?

[View all Related Programs](#)

## 2. What is an Array?

- An [array](#) is commonly referred to as a collection of items stored at contiguous memory locations.
- Items stored are of the same type.
- It organizes data so that a related set of values can be easily sorted or searched.

Fig: Array

## 3. What is a Linked List?

- Like an array, a [linked list](#) refers to a linear data structure in which the elements are not necessarily stored in a contiguous manner.
- It is basically a sequence of nodes, each node points towards the next node forming a chain-like structure.

Fig: Linked List

## 4. What is LIFO?



- LIFO IS AN APPREVIATION FOR LAST IN FIRST OUT

- It is a way of accessing, storing and retrieving data.
- It extracts the data that was stored last first.

## 5. What is a Stack?

- A stack refers to a linear data structure performing operations in a LIFO (Last In First Out) order.
- In a stack, elements can only be accessed, starting from the topmost to the bottom element.

## 6. What is FIFO?

- FIFO stands for First In First Out.
- It is a way of accessing, storing and retrieving data.
- The data that was stored first is extracted first.



Fig: LIFO, FIFO

Till now, you've covered some very fundamental coding interview questions. Going ahead you will dive deeper into the subject.

## 7. What is a Queue?

- A **queue** refers to a linear data structure that performs operations in a FIFO order.
- In a queue, the least recently added elements are removed first as opposed to a stack.



Fig: Queue

## 8. What are Binary Trees?

- A binary tree is an extension of the linked list structure where each node has at most two children.
- A binary tree has two nodes at all times, a left node and a right node.



## Fig: Binary Trees

### 9. What is Recursion?

- **Recursion** refers to a function calling itself based on a terminating condition.
- It uses LIFO and therefore makes use of the stack data structure.

The next couple of coding interview questions will explore your knowledge of OOPs.

### 10. What is the OOPs concept?

OOPs stands for **Object-Oriented Programming** System, a paradigm that provides concepts such as objects, classes, and inheritance.

### 11. What are the concepts introduced in OOPs?

Following are the concepts introduced in OOPs:

- Object - A real-world entity having a particular state and behavior. We can define it as an instance of a class.
- Class - A logical entity that defines the blueprint from which an object can be created or instantiated.
- Inheritance - A concept that refers to an object gaining all the properties and behaviors of a parent object. It provides code reusability.
- Polymorphism - A concept that allows a task to be performed in different ways. In Java, we use method overloading and method overriding to achieve polymorphism.
- Abstraction - A concept that hides the internal details of an application and only shows the functionality. In Java, we use abstract class and interface to achieve abstraction.
- Encapsulation - A concept that refers to the wrapping of code and data together into a single unit.

This is one of the very common coding interview questions, that often allows the interviewer to branch out into related topics based on the candidate's answers

### 12 Explain what a Binary Search Tree is

- A [binary search tree](#) is used to store data in a manner that it can be retrieved very efficiently.
- The left sub-tree contains nodes whose keys are less than the node's key value.
- The right sub-tree contains nodes whose keys are greater than or equal to the node's key value

Fig: Binary Search Tree

## Become a Software Development Professional



### Full Stack Java Developer

Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts

Hands-on practice through 20+ projects, assessments, and tests

6 months

[View Program](#)



### Full Stack Web Developer - MEAN Stack

Comprehensive Blended Learning program

8X higher interaction in live online classes conducted by industry experts

11 months

[View Program](#)

Not sure what you're looking for?

[View all Related Programs](#)

## 13. Explain Doubly Linked Lists?

- [Doubly linked lists](#) are categorized as a special type of linked list in which traversal across the data elements can be done in both directions.
- This is made possible by the presence of two links in every node, one that links to the node next to it and another that connects to the node before it.

### Fig: Doubly Linked List

#### 14. What is a Graph?

- A graph is a particular type of data structure that contains a set of ordered pairs.
- The ordered pairs in a graph are also known as edges or arcs and are most commonly used to connect nodes where the data can be stored and retrieved.

#### 15. Differentiate between linear and non-linear data structure?

Linear data structure	Non-linear data structure
It is a structure in which data elements are adjacent to each other	It is a structure in which each data element can connect to over two adjacent data elements
Examples of linear data structure include linked lists, arrays, queues, and stacks	Examples of nonlinear data structure include graphs and trees

#### 16. What is a Deque?

- A deque is a double-ended queue.
- This is a structure in which elements can be inserted or removed from either end.

#### 17. What's the difference between Stack and Array?

Stack	Array
Stack follows a Last In First Out (LIFO) pattern. What this means is that data access necessarily follows a particular sequence where the last data to be stored is the first one that will be extracted.	On the other hand, Arrays do not follow a specific order, but instead can be accessed or called by referring to the indexed element within the array.

## 18. Which sorting algorithm is the best?

- There are many types of sorting algorithms: bubble sort, quick sort, balloon sort, merge sort, radix sort, and more.
- No algorithm can be considered as the best or fastest because they have designed each for a specific type of data structure where it performs the best

## 19. How does variable declaration affect memory?

- The amount of memory that is to be reserved or allocated depends on the data type being stored in that variable.
- For example, if a variable is declared to be “integer type”, 32 bits of memory storage will then be reserved for that particular variable.

## 20. What are dynamic data structures?

Dynamic data structures have the feature where they expand and contract as a program runs. It provides a very flexible method of data manipulation because adjusts based on the size of the data to be manipulated.

These 20 coding interview questions that test the conceptual understanding of the candidates give the interview a clear idea on how strong the candidate's fundamentals are

## Programming Interview Questions

The next set of coding interview questions focus tests the [programming](#) expertise of the candidates and dives deep into various related aspects.

The code screenshots given along with the below coding interview questions helps you provide the answer to the question, with clarity.

## 21. How do you reverse a string in Java?

- Declare a string

- Take out the length of that string.
- Loop through the characters of the string.
- Add the characters in reverse order in the new string.

```
String str = "hello";  
  
String reverse = "";  
  
int length = str.length();  
  
for (int i = 0; i < length; i++) {  
  
    reverse = str.charAt(i) + reverse;  
  
}  
  
System.out.println(reverse);
```

## 22. How do you determine if a string is a palindrome?

- A string is a palindrome when it stays the same on reversing the order of characters in that string.
- It can be achieved by reversing the original string first and then checking if the reversed string is equal to the original string.

```
if (str.equals(reverse)) {  
  
    System.out.println("Palindrome");  
  
} else {  
  
    System.out.println("Not Palindrome");  
  
}
```

## 23. Find the number of occurrences of a character in a String?

To find the number of occurrences, loop through the string and search for that character at every iteration; whenever it is found, it will update the count.

```
int count = 0;

char search = 'a';

for (int i = 0; i < length; i++) {

    if (str.charAt(i) == search) {

        count++;

    }

}

System.out.println(count);
```

## 24. How to find out if the given two strings are anagrams or not?

Two strings are anagrams if they contain a similar group of characters in a varied sequence.

- Declare a boolean variable that tells at the end of the two strings are anagrams or not.
- First, check if the length of both strings is the same, if not, they cannot be anagrams.
- Convert both the strings to character arrays and then sort them.
- Check if the sorted arrays are equal. If they are equal, print anagrams, otherwise not anagrams.

```
boolean anagramstat = false;

if (str.length() != reverse.length()) {

    System.out.println(str + " and " + reverse + " not anagrams string");

} else {

    char[] anagram1 = str.toCharArray();
```

```
char[] anagram2 = reverse.toCharArray();

Arrays.sort(anagram1);

Arrays.sort(anagram2);

anagramstat = Arrays.equals(anagram1, anagram2);

}

if (anagramstat == true) {

    System.out.println(" anagrams string");

} else {

    System.out.println(" not anagrams string");

}
```

## 25. How do you calculate the number of vowels and consonants in a String?

- Loop through the string.
- Increase the vowel variable by one whenever the character is found to be a vowel, using the if condition. Otherwise, increment the consonant variable.
- Print the values of both the vowel and the consonant count.

```
int vowels = 0;

int consonants = 0;

for (int k = 0; k < str.length(); k++) {

    char c = str.charAt(k);

    if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')

        vowels++;

}
```

```
else
    consonants++;
}

System.out.println("Vowel count is " + vowels);

System.out.println("Consonant count is: " + consonants);
```

## 26. How do you get the matching elements in an integer array?

- Declare an array.
- Nest a couple of loops to compare the numbers with other numbers in the array.
- Print the matching elements if found.

```
int[] a = { 1, 2, 3, 4, 5, 1, 2, 6, 7 };

for (int m = 0; m < a.length; m++) {
    for (int n = m + 1; n < a.length; n++) {
        if (a[m] == a[n])
            System.out.print(a[m]);
    }
}
```

## 27. How would you implement the bubble sort algorithm?

- Declare an array.
- Nest a couple of loops to compare the numbers in the array.
- The array will be sorted in ascending order by replacing the elements if found in any other order.

```
int[] a = { 1, 2, 7, 6, 4, 9, 12 };
```

```
for (int k = 0; k < a.length; k++) {  
  
    for (int l = 0; l < a.length - l - 1; l++) {  
  
        if (a[l] > a[l + 1]) {  
  
            int t = a[l];  
  
            a[l] = a[l + 1];  
  
            a[l + 1] = t;  
  
        }  
  
    }  
  
}
```

Learn the top skills in-demand including Angular, Spring Boot, JSPs, and SOA to build highly web scalable apps with the [Full Stack Java Developer Masters Program](#).

## 28. How would you implement the insertion sort algorithm?

- We assume the first element in the array to be sorted. The second element is stored separately in the key. This sorts the first two elements. You can then take the third element and do a comparison with the ones on the left of it. This process will go on until a point where we sort the array.

```
int[] a = { 1, 2, 7, 6, 4, 9, 12 };  
  
for (int m = 1; m < a.length; m++) {  
  
    int n = m;  
  
    while (n > 0 && a[n - 1] > a[n]) {  
  
        int k = a[n];  
  
        a[n] = a[n - 1];  
  
    }
```



```
a[n - 1] = k;
```

```
n--;
```

```
}
```

```
}
```

## 29. How do you reverse an array?

- Loop till the half-length of the array.
- Replace the numbers corresponding to the indexes from the starting and the end.

```
int[] a = { 1, 2, 7, 6, 4, 9, 12 };
```

```
for (int t = 0; t < a.length / 2; t++) {
```

```
    int tmp = a[t];
```

```
    a[t] = a[a.length - t - 1];
```

```
    a[a.length - t - 1] = tmp;
```

```
}
```

## Become a Software Development Professional



### Full Stack Java Developer

Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts

Hands-on practice through 20+ projects, assessments, and tests

6 months

[View Program](#)



### Full Stack Web Developer - MEAN Stack

Comprehensive Blended Learning program

8X higher interaction in live online classes conducted by industry experts

11 months

[View Program](#)

Not sure what you're looking for?

[View all Related Programs](#)

### 30. How would you swap two numbers without using a third variable?

- Declare two variables and initialize them with values.
- Make b the sum of both numbers.
- Then subtract the sum (b) from a, so a is now swapped.
- Lastly, subtract a from the sum (b), so b is also swapped.

```
int a = 10;
```

```
int b = 20;
```

```
b = b + a; // now b is sum of both the numbers
```

```
a = b - a; // b - a = (b + a) - a = b (a is swapped)
```

```
b = b - a; // (b + a) - b = a (b is swapped)
```

### 31. Print a Fibonacci series using recursion?

- The [Fibonacci](#) numbers are the numbers in the following integer sequence:  
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, .....
- We can calculate them using the mathematical formula used in the [Fibonacci recursive function](#).

```
public static int fibonacci(int n) {
```

```
    if (n <= 1)
```

```
        return n;
```

```
    return fibonaci(n - 1) + fibonaci(n - 2);
```

```
}
```

```
public static void main(String args[]) {
```

```
    int n = 10;
```

```
    System.out.println(fibonacci(n));
```

```
}
```

### 32. How do you find the factorial of an integer?

- A factorial is a function that multiplies a number by every number below it. For example,  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ .
- Recursive function multiples the numbers until it reaches 1.

```
public static long factorial(long n) {
```

```
    if (n == 1)
```

```
        return 1;
```

```
    else
```

```
        return (n * factorial(n - 1));
```

```
}
```

### 33. How do you reverse a Linked List?

- Declare a linked list.
- Add elements to that linked list.
- Apply the descending iterator method to the linked list.
- This reverses the order of elements in the linked list.

```
LinkedList<Integer> ll = new LinkedList<>();
```



```
ll.add(1);

ll.add(2);

ll.add(3);

System.out.println(ll);

LinkedList<Integer> ll1 = new LinkedList<>();

ll.descendingIterator().forEachRemaining(ll1::add);

System.out.println(ll1);
```

### 34. How would you implement Binary Search?

- Binary search divides the array into half in every iteration step until it finds the element.
- It works on the sorted arrays since it compares the values of adjacent elements and then calculates the mid number.
- If the value of low becomes greater than high at any point, it means the element is not present in the list.

```
int mid = (low + high) / 2;

while (low <= high) {

    if (arr[mid] < key) {

        low = mid + 1;

    } else if (arr[mid] == key) {

        return mid;

    } else {
```

```
        high = mid - 1;
```

```
}
```

```
    mid = (low + high) / 2;
```

```
}
```

```
if (low > high) {
```

```
    return -1;
```

```
}
```

```
return -1;
```

### 35. How would you find the second largest number in an array?

- Loop through the array.
- If the value of i is greater than the highest, store the value of i in highest, and store the value of highest in the second-highest variable.

```
private static int findSecondHighest(int[] array) {
```

```
    int highest = Integer.MIN_VALUE;
```

```
    int secondHighest = Integer.MIN_VALUE;
```

```
    for (int i : array) {
```

```
        if (i > highest) {
```

```
            secondHighest = highest;
```

```
            highest = i;
```

```
        } else if (i > secondHighest) {
```

```
            secondHighest = i;
```

```
}
```

```
}
```

```
    return secondHighest;
```

```
}
```

### 36. How do you remove all occurrences of a given character from the input string?

- Use the built-in string method “replace” to replace a character with any other character, including symbols and white spaces.

```
String str1 = "Australia";
```

```
str1 = str1.replace("a", "");
```

```
System.out.println(str1); //ustrli
```

### 37. Showcase Inheritance with the help of a program?

- The class Cat inherits the property color from the class Animal by extending the parent class (Animal).
- This way a class Cat can have more parent classes if it wishes to inherit their properties.

```
class Animal {
```

```
    String color;
```

```
}
```

```
class Cat extends Animal {
```

```
    void meow() {
```

```
        System.out.println("Meow");
```

```
}
```

```
}
```

### 38. Explain overloading and overriding with the help of a program?

Overloading:

<https://www.simplilearn.com/coding-interview-questions-article>

20/26

When a class has two or more methods with the same name, they are called overloaded methods.

```
class Foo {  
  
    void print(String s) {  
  
        System.out.println(s);  
  
    }  
  
    void print(String s, int count) {  
  
        while (count > 0) {  
  
            System.out.println(s);  
  
            count--;  
  
        }  
  
    }  
}
```

Overriding:

When a superclass method is also implemented in the child class, it's a case of overriding.

```
class Base {  
  
    void printName() {  
  
        System.out.println("Base Class");  
  
    }  
}
```

```
class Child extends Base {
```

```
    @Override
```

```
    void printName() {
```

```
        System.out.println("Child Class");
```

```
}
```

```
}
```

## Become a Software Development Professional



### Full Stack Java Developer

Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts

Hands-on practice through 20+ projects, assessments, and tests

6 months

[View Program](#)



### Full Stack Web Developer - MEAN Stack

Comprehensive Blended Learning program

8X higher interaction in live online classes conducted by industry experts

11 months

[View Program](#)

Not sure what you're looking for?

[View all Related Programs](#)

## 39. How do you check if the given number is prime?

- Use if statements to check for each condition separately:
  - If the number is 0 or 1, it cannot be prime.
  - If the number is 2, it is prime number.
  - If the number is divisible by other numbers, it is prime.

• If the number is divisible by other numbers, it is prime.

```
public static boolean isPrime(int n) {  
  
    if (n == 0 || n == 1) {  
  
        return false;  
  
    }  
  
    if (n == 2) {  
  
        return true;  
  
    }  
  
    for (int i = 2; i <= n / 2; i++) {  
  
        if (n % i == 0) {  
  
            return false;  
  
        }  
  
    }  
  
    return true;  
  
}
```

#### 40. How do you sum all the elements in an array?

- Use for loop to iterate through the array and keep adding the elements in that array.

```
int[] array = { 1, 2, 3, 4, 5 };
```

```
int sum = 0;
```

```
for (int i : array)
```

```
    sum += i;
```

```
System.out.println(sum);
```

As you get prepared for your job interview, we hope that these Coding Interview Questions have provided more insight into what types of questions you are likely to come across.

## Choose The Right Software Development Program

This table compares various courses offered by Simplilearn, based on several key features and details. The table provides an overview of the courses' duration, skills you will learn, additional benefits, among other important factors, to help learners make an informed decision about which course best suits their needs.

Program Name	Full Stack Java Developer	Automation Testing Masters
Geo	IN	All
University	Simplilearn	Simplilearn
Course Duration	6 Months	11 Months
Coding Experience Required	Basic Knowledge	Basic Knowledge
Skills You Will Learn	15+ Skills Including Core Java, SQL, AWS, ReactJS, etc.	Java, AWS, API Testing, -
Additional Benefits	Interview Preparation Exclusive Job Portal 200+ Hiring Partners	Structured Guidance Learn From Experts Hands-on Training
Cost	\$\$	\$\$
	<a href="#">Explore Program</a>	<a href="#">Explore Program</a>

# 30 Basic Job Interview Questions & How To Answer Them

## 1. Can you tell me more about yourself?

“Tell me about yourself” should be [expected in job interviews](#), but talking about yourself can feel uncomfortable. It’s often asked as a first question or an icebreaker. Your answer gives the interviewer a sense of who you are and your background beyond your resume. When answering this question, **focus on experiences that highlight your qualifications.**

It’s OK to mention personal interests that will [help the hiring manager connect with you](#), such as a shared alma mater or a [nonprofit interest](#). But for the most part, you should steer your response toward how you’ll contribute to the organization. Gather specific skills and accomplishments that make you a strong candidate. Take the opportunity to start the interview off with points that paint a positive picture.

### Example Answer:

*I’m a recent college graduate with a degree as a certified accountant from Kentucky State University. I also completed an internship at a financial management firm, where I got hands-on experience in financial analysis. During my time at the firm, I discovered that I enjoyed managing finances and creating investment plans for retirees, which led me to apply for this role.*

## 2. What do you think your greatest weakness is?

Every candidate puts their best foot forward during the hiring process. Hiring managers are looking for a glimpse of your self-awareness and how you [manage constructive criticism](#). When answering this question, it’s essential to **be honest but frame your response positively**. Phrase your answer in a way that shows you’re actively seeking personal and [career growth and development](#).

### Example Answer:

*I think my greatest weakness is that I struggle a bit with balancing multiple priorities. I can lose myself in a project, which can make me feel overwhelmed and stressed when numerous tasks are on my to-do list. To manage this, I’ve*

*been setting smaller goals and creating better task management and daily workflow routines.*

### 3. What do you feel your greatest strength is?

The hiring manager is curious about **how well you know yourself** and why you think you'll excel in this role. This is a great opportunity to **show how passionate you are about the job** and explain why your experience makes you a great fit.

#### Example Answer:

*I'm incredibly excited about this role because I'm passionate about seeing the big picture and finding more efficient workflows. For example, in my previous role, I identified opportunities to streamline a simple daily process in how tasks were communicated and completed. Ultimately, those tiny savings added up, and they saved the company an hour per day, which added up to savings of over 250 hours of labor yearly. This role is an excellent fit for me because it allows me to use those same analytical and problem-solving skills within a larger organization.*

### 4. Why are you interested in this role?

Hiring managers are looking to discover how well you **researched the company**. Are you looking for any open position, or are you excited about the opportunities this particular role and organization offer? Here, you can **share specifics about the company and why this job is the perfect fit for your career goals**.

#### Example Answer:

*I'm interested in this role because it offers me the chance to combine my marketing knowledge with my passion for helping others. I learned about the organization's mission and how it helps people improve their lives through education and financial literacy. I've watched the jobs page for roles fitting my skill set, and I was delighted when this position was posted. I know I can be an asset to the team, and I'm excited about this role's potential opportunities.*

### 5. Where do you see yourself in five years?

Hiring and turnover are costly. When you're applying, responsible leaders are looking to **ensure that your goals and plans align with the company's visions and**

**career paths.** The best way to answer this question is to clearly understand your **long-term development goals** and how this position can help you achieve them.

### Example Answer:

*In five years, I hope to become an industry expert that inspires school-aged kids from all backgrounds to pursue an **engineering** career. Ideally, I'd love to use this role as a platform to learn and grow to become a company leader where I can mentor new team members. As I mentioned previously, the company's culture and mission statement align closely with my values, and in five years, I'm hoping to be here, growing my career.*

*Ready to land your next interview? Connect with **companies hiring now!***

## 6. Why are you looking for a new role?

Don't air all of your dirty laundry here. You want to steer away from focusing on any negative aspects of your current role. Instead, **focus on why you're excited about the new job and how it fits into your career path.** Compare the two companies if you can find a way to do so that doesn't sound like you're throwing your current company under the bus.

### Example Answer:

*I'm looking for a new role because I want to pursue an opportunity that will help me expand my skill set and build upon what I've learned so far. The company I work for has a well-established senior team, so there's less room for growth. One of the things I love about my current job is the ability to impact small business success with efficient web design. This position is the perfect combination. I'll have the chance to challenge myself in a new way while working more closely with the customer to bring their vision to life.*

## 7. What motivates you on a daily basis?

Employers are looking to **ensure that your intrinsic motivations align with long-term company values and goals.** If **your motivation** is a specific benefit or salary range, that will wear off at some point, and your job performance will suffer. On the other hand, if you're passionate about the mission or find the duties creative and challenging, you're more likely to be a consistently high-performing employee.

**Example Answer:**

*As a care advocate, I am passionate about helping patients navigate the complicated insurance and medical system. On a personal level, I've seen the devastating impact that occurs when patients postpone routine care. I look forward to the ability to personally support others' health in positive ways.*

**8. Why are you the best candidate for this role?**

This question can feel like you're walking a tightrope. You don't know the other candidates and don't want to sound arrogant by saying you're the best. But on the other hand, you need to **communicate your confidence and job fit**. The solution is rooted in your research and **reframing your response into one that shows you're uniquely qualified**. You can tie in previous experiences and skill sets with the job duties to highlight your excellent fit.

**Example Answer:**

*What makes me uniquely qualified for the role is my experience leading a high-performing sales team at a larger agency. We consistently led the company and exceeded our goals. Not only that, but we had exceptional customer service reviews with a high ratio of returning customers. Blending that experience with my time spent mentoring new team members makes me an excellent fit for this role.*

**9. What three words would your most recent teammates use to describe you?**

Even when a role is relatively independent, you'll still need to **interact and collaborate with others**. Hiring managers are looking to discover **how well you work within the boundaries of a team**. Consider the traits and skills you've been praised for and feedback you've received during performance reviews.

**Example Answer:**

*My current teammates would describe me as reliable, adaptable, and proactive. I understand that meeting my personal goals can't get me very far without us succeeding as a team. Because of that, my teammates' feedback is consistently that I am an excellent team player who takes initiative and is open to feedback.*

## 10. Describe your ideal company culture.

In recent years, U.S. companies have [spent over \\$223 billion due to employee turnover](#). Much of that stems from a poor cultural fit. Even if the organization doesn't have [a toxic culture](#), ensuring applicants mesh with the team and thrive personally helps lower that number substantially. Not only that, but if they have an excellent reputation and a unique culture, they will be rigorous in defending it. Do your research and **be specific in explaining why you'll enhance the company culture.**

### Example Answer:

*In my experience, I've thrived at companies that balance team goals with opportunities for personal development. I'm excited about working in a smaller team with a diverse group. I believe encouraging employees to bring their unique perspectives and share ideas will lead to creative solutions. I also look forward to learning from teammates with different backgrounds, experiences, and skill sets. Ultimately, my ideal company culture values collaboration, creativity, and learning from diverse perspectives.*

*Ready to land your next interview? Connect with [companies hiring now!](#)*

## 11. What would you do in your first 90 days in this role?

Will you be able to hit the ground running? **Hiring managers want to know that you can strategize and plan.** When discussing goals, focus on topics such as integrating yourself with the team, developing relationships with other departments, and learning more about the role. Keep it specific by describing projects or tasks you think you'll prioritize within those 90 days.

### Example Answer:

*In my first 90 days in this role, I will focus on learning more about the team and our unique contribution to the industry. I plan to meet with key stakeholders, review current projects, understand our process flow, and build relationships with other departments. Additionally, I think it's essential to gain a thorough understanding of our product offering so that I can support our customers and ensure we meet their needs. As part of this, I'd like to review*

*customer feedback and develop a plan to ensure I'm consistently exceeding their expectations.*

## 12. Give an example of when you received negative feedback and how you responded.

Hiring managers are seeking to understand whether or not you have a growth mindset. Negative feedback is inevitable, and leaders are looking for those who understand that criticism isn't personal. Instead, you want to **communicate that you'll use feedback to improve your skills**. Use [the STAR method](#) to describe a situation, the feedback you received, and how you applied it to achieve a better outcome.

### Example Answer:

*Recently, I was given negative feedback about a large project I completed as a virtual assistant. My supervisor noted that the formatting of my presentation needed to be in line with company standards and clarified for the audience. I took this feedback and spent several days revising the presentation by making sure all elements were cohesive and easy to understand. I also contacted my supervisor to ensure that we updated the style guide to clarify their standards moving forward. In the end, they were happy with the project and appreciated that I took the initiative to improve the document for everyone on the team.*

## 13. Describe a problem you solved in a creative way.

Creative thinking is a tremendous asset in today's workforce. Business continues to get more competitive, and thinking outside the box can ensure the company stays on the leading edge of the industry. Consider the role you're applying for and **highlight a time you solved a problem the position would have to deal with**.

### Example Answer:

*As a marketing manager, I was tasked with increasing our online presence without increasing our budget. I identified an opportunity to leverage influencers in a few other industries who had an engaged and growing audience. By forming creative partnerships with these influencers, we*

*increased our brand's visibility and drove more traffic to our website without investing in ad spending.*

#### 14. Describe a time when you had to make a decision without time to research.

When you're interviewing, **hiring managers often want to understand your ability to think on your feet**. Especially **in leadership roles**, you'll need to use the information you have at the time to make the best decision possible. Depending on the situation, you can touch on how you'd make a different decision in the future now that you have more information.

##### **Example Answer:**

*When I was managing a hospitality team, a time-sensitive issue with a customer's room cleanliness needed to be addressed due to their travel constraints. I had to decide on the best course of action, as the housekeeping manager needed to be available to gather more information. After brief consideration, I offered them a refund, which wasn't policy. After they left, we determined that they were placed in a room that hadn't been cleaned due to a computer error. As such, it was the right course of action. Regardless, it's always best to use the available information to give the best possible customer service for the situation.*

#### 15. Why are you leaving your current job?

No matter how toxic your current work environment is, it will still reflect poorly on you if you focus on the negative. Instead, you can **talk about how you're looking for new challenges or a better fit for your skill set**, but **avoid being hostile toward your current workplace or colleagues**. Even if there's drama or you have a horrible leader, you'll want to reframe your answers positively.

##### **Example Answer:**

*I'm currently at a point in my career where I want to take on new challenges and grow my skill set further. My current role has been a great learning experience, but I'm intrigued by the opportunity that this position provides to utilize my expertise differently. I'm eager to expand my knowledge and grow professionally, which is why I'm excited about this opportunity.*

**Ready to land your next interview? Connect with companies hiring now!**

## 16. Describe a time when you went above and beyond for a customer or client.

Showing you can take the initiative and are willing to go the extra mile speaks volumes about your character and professionalism. When describing a time you went above and beyond, try to **focus on what drove you to go above and beyond**. Was it a concern for the customer? A desire to ensure everything went smoothly?

### **Example Answer:**

*As an event planner, I had a customer with incredibly high expectations for their event. Because of the complexity of their requests and timeline, I knew it would be challenging, but I could also see that they felt pressured to put on a memorable event. To ensure everything went smoothly, I put in extra time and effort beforehand to create a detailed plan and ensure the staff was fully prepared. I also kept the customer updated on the status of various details. That required extra communication and time, but it helped them relax and feel confident in the outcome. My hard work paid off, as the customer was delighted with how everything turned out. It was a great feeling to have exceeded their expectations.*

## 17. Why do you want to work remotely?

Remote work can be a bit glorified in the media and modern culture. Hiring managers want to ensure **you have realistic expectations** and a plan to address the challenges. **Keep your answer focused on the professional benefits**. Explain how working remotely can help you meet your development goals and would work well with your personality type.

### **Example Answer:**

*I'm a self-starter motivated to stay productive and reach my goals. I find that having the flexibility to work from different locations increases my productivity, allowing me to avoid office distractions and politics. Additionally, working remotely allows me to work in a custom-tailored environment to boost my productivity. I'm also looking forward to using the*

*time I'm saving by not commuting to fit in more exercise, which boosts my creativity.*

## 18. How will you handle or minimize distractions when you work remotely?

Recruiters are hesitant to hire remote workers who aren't realistic about the distractions you'll face working and living in the same location. It is essential to **show the interviewer that you can work productively even when you are not in a traditional office setting.**

An excellent way to do this is by outlining a few strategies that help you stay on track and focused. You can share things like setting up specific hours for working, turning off notifications and other distractions, and setting boundaries for yourself and others.

### Example Answer:

*I realize it can be easy to get distracted by the things around you, especially when you're at home. To combat this, I plan to use a few strategies that help me stay focused and on track. First, I'll set up and post specific hours for my workday next to my desk. That way, my family will know when I'm available and when I need to focus. Second, I'll turn off all notifications and distractions to stay focused. Finally, I will be diligent about prioritizing my tasks to ensure I stay on top of my deadlines. By using these strategies, I'll be able to maximize my productivity.*

## 19. How do you stay connected to your teammates when you don't work in the same location?

Despite working alone, leaders need team members who **facilitate communication and collaboration** despite distances and varying time zones. You'll need to **demonstrate that you're knowledgeable, capable of using communication tools, and driven** to reach out.

### Example Answer:

*I'm very familiar with collaboration platforms, such as Zoom and Slack, and I'm comfortable participating in virtual meetings and team projects. I also actively seek time for informal conversations, so I schedule weekly coffee or lunch sessions with my colleagues. That way, we can catch up, have*

*meaningful conversations, and brainstorm ideas. I respect everyone's communication preferences and give them space to respond as they need.*

## 20. How do you set boundaries between your home life and your work tasks while working remotely?

Working from your home office can have many perks. But it also comes with a higher risk of burnout if you're not intentional about setting boundaries. It's all too easy to slip into a mode where you're always working. You need to **demonstrate that you are capable of establishing and maintaining boundaries.**

### Example Answer:

*I know that it can be challenging to separate home life from work when working remotely, so I take proactive steps to ensure I'm not working around the clock. To do this, I have a dedicated office space, set specific hours for my workday, and maintain a routine that helps me shift from work mode to home mode. During my off-work hours, I turn off any notifications related to work and disconnect from my computer. Additionally, I take regular breaks throughout the day and steer clear of eating lunch at my desk.*

*Ready to land your next interview? Connect with **companies hiring now!***

## 21. Do you consider yourself a self-starter?

Without a manager stopping by your desk, you'll need the ability and motivation to take charge of your workload. Working remotely, **leaders will be relying on you to take the initiative** to **manage your projects** and stay on top of your deadlines.

Share traits that reinforce that you're a self-starter.

### Example Answer:

*I've always had an entrepreneurial spirit and am highly motivated to complete tasks independently. I enjoy challenging myself to complete tasks by a specific time or with a set outcome. I'm also highly organized and can prioritize my work tasks to finish everything on time.*

## 22. What does a typical day look like for you? What is your daily routine when you work remotely?

It might not seem vital, but your daily routine can make or break your success while working from home. Without social cues, you can quickly develop bad habits that **lead to burnout** and significantly impact your wellness. Outline any routines that **demonstrate how you prioritize and manage your tasks while working remotely.**

**Example Answer:**

*My day starts with a cup of tea and a short workout to get my head in the best space. After that, I maintain my morning routine the same way I would if I worked in an office. I clock in at my set hours, check my emails, and review any new daily tasks. I use a planner and goal-setting sheet to set my daily goals. Using the Pomodoro Technique, I work in increments that allow me to stay focused and alert, taking small breaks every hour or so. When the day is done, I log off and take time for myself and my family. This helps me reset my mind and prepare for the next day. Overall, having a daily routine has been highly beneficial in helping me stay productive and organized when working remotely. It allows for sufficient breaks, so I stay energized and motivated.*

**23. How have you managed to stay productive while working from home?**

When your team isn't in the same location, it can be challenging to stay productive. From distractions to technical difficulties, plenty of things can get in the way of your workflow. **Share specific examples of how you protect your focus despite distractions.**

**Example Answer:**

*Admittedly, I learned the hard way that working from home isn't easy if you don't have a routine set in place or the right tools to stay motivated. Working remotely requires a certain level of self-discipline, and I initially struggled to avoid distractions. Recognizing that I wasn't thriving, I put new workflows in place to keep myself on track. I practice good time management skills, set realistic daily goals, and use task lists to prioritize my work. Also, I've scheduled a routine check-in with my manager, which helps me stay connected and in the loop on changing priorities.*

**24. What is your expected salary?**

Beyond wondering if they can afford you, **interviewers want to see if you've thoroughly researched the industry and the position.** If you don't value yourself and your work, that might signal a lack of confidence in your performance. It's tricky since duties can vary between roles and geography. If you still need to learn all the specifics, you can either give a salary range based on research or answer that you'd like more information before **negotiating specific salary and benefits.**

#### **Example Answer:**

*After researching the industry and taking my experience and education into account, I'm targeting a salary between \$50,000-\$60,000. However, I'd like to learn more about the position's specific duties and benefit offerings before we discuss a fair amount for the role.*

#### **25. Can you talk me through your career gap, the reasons you left the workforce, and how you feel it's affected you professionally?**

If you were out of the workforce for extended periods, it's essential that you can explain why and how it impacted your career. You'll want to **show that your gap was due to intentional and responsible decision-making**, rather than an inability to focus or commit to a career path. Tie in professional growth, such as **volunteer or freelance work**, that you completed during your gap.

#### **Example Answer:**

*I was out of the workforce for five years while raising a family after graduating college. With my partner in the military, we decided that the timing was best for our careers and families as we anticipated several moves. Now that my partner has left active duty and our children are a bit older, we're ready to move to the next stage. I am quite thankful I had that time to focus on volunteer work and exploring different career paths related to my degree. With some hands-on work, I discovered my passions, which led me to this role.*

*Ready to land your next interview? Connect with **companies hiring now!***

#### **26. Can you share a time you had a conflict with a teammate and how you handled it?**

You've inevitably had a difference of opinion with someone in the workforce. It's essential to show that you can **handle it skillfully** and professionally without

devaluing other ideas. **Interviewers want to see how you approach difficult conversations and your ability to defuse tensions** while maintaining professional composure.

#### **Example Answer:**

*I recently had a conflict with my teammate over an upcoming project. He wanted to tackle it with a method that I felt was outdated. We discussed our ideas at length. I could see that he didn't understand my vision, so I found a new way to explain it and shared some similar examples. With some additional explanation, he opened up to my suggestions, and we were able to blend both of our approaches for the best outcome.*

#### **27. What skills from your previous industry do you feel translate to this position?**

When you're looking to change careers, you'll need to connect the dots for the hiring manager. They'll often need you to **explain how your skills and experience translate to the new role or field**, especially if they're not familiar with your previous duties. Highlight ways you can bring a fresh perspective while drawing on aspects of your experience.

#### **Example Answer:**

*In my previous role as an account manager, I worked closely with customers to identify their needs and offered solutions. This translates well into this HR position because it's a role that supports our internal customers, just in a different way. Additionally, the data analysis skills I gained while working with customer feedback have been invaluable in creating reports on employee satisfaction. Furthermore, my familiarity with identifying trends and developing solutions to business challenges can benefit the team in various ways. Overall, I believe many of my skills will transfer to this role and fill any gaps.*

#### **28. Why are you looking to leave your current industry?**

When you're transitioning to a new career, it's essential that you come across as motivated, rather than desperate. You want to make sure that the interviewer understands that you're eagerly moving toward something, rather than simply

escaping a poor job fit. You'll need to **reassure the interviewer that your move is intentional because you think you'll thrive** instead of moving blindly away from your previous career.

#### **Example Answer:**

*I'm looking to move away from my current industry because I've achieved all I can. I'm passionate about my company, and my role has been enriching. However, I've reached a ceiling with my current position and need to find an opportunity that will challenge me further. That's why this new position is so attractive to me—it offers an exciting chance to learn new skills and use the experience I've gained to develop further. Plus, I believe it will allow me to grow professionally and personally in ways that my current job doesn't provide.*

#### **29. Describe your ideal work environment.**

Your interviewer wants to know if you'll **fit their corporate culture** well. Use your research into the company culture and notes from the job posting. But be cautious that you're not simply quoting what you read. Instead, **reframe culture cues as personal motivations.**

#### **Example Answer:**

*I thrive in an environment that encourages learning and ideas. I'm always looking for new challenges, so having a goal-oriented workplace where everyone works together to meet deadlines and service targets is essential. My **ideal work environment** also values collaboration and open communication. This is something that lays the foundation for success as we can come up with better solutions, rather than aggressively competing. At the end of the day, the sales numbers don't matter if we lose customers due to toxic competitiveness.*

#### **30. What questions do you have for me?**

It may seem like this is just a bonus question that might not matter that much, but it's actually an essential part of the interview. **Your questions will show how engaged you are in the role**, how much research you've done, and of course, if there's anything else you want to know. This is your chance to **impress the hiring**

manager and ensure they remember you. Of course, your question will depend a bit on what's already been covered.

### Example Answers:

*I'm interested in learning more about what success looks like in this role. Do you have any clear objectives or metrics I could use to measure progress?*

*What are your team's biggest challenges, and how can someone in this role help overcome them?*

*I saw from your LinkedIn profile that you've been with the company for over 10 years. What culture attributes have inspired you to stay with the company?*

**Ready to land your next interview? Connect with companies hiring now!**

## Basic Interview Questions

It's nearly impossible to prepare completely for each question, as there are endless ways a hiring manager can word their inquiries. But the majority of questions fit into a few different categories.

Think through your experiences and **prepare to discuss situations that highlight your team and culture fit, your relevant skills, and your ability to apply essential characteristics**. You may need to reframe them to fit the exact phrasing, but you'll be ready for just about anything if you start with these questions and then expand to some broader experiences you can reference.

And if you find that this isn't your dream role, we can help you find one that is. No matter the role, no matter the schedule, the FlexJobs database of remote and flexible jobs has something that works for you. Members get full access to our jobs database and so much more. Take the tour to learn more about all of the ways a FlexJobs membership can support your job search.

[Take the Tour](#)

Don't forget to share this article with friends!



through all the common interview c

[Register](#)[Sign In](#)

2. Checking out the situational interview questions section and learning how to answer questions that are relevant for you
3. Learning what's the idea behind behavioral interview questions, so you're prepared to answer whatever the HR manager shoots at you

## From Awkward to Awesome

### How to Nail Job Interview Questions

 novoresume

## Most Common Job Interview Questions:

1. Tell me something about yourself.
2. How did you hear about this position?
3. Why do you want to work here?
4. Why did you decide to apply for this position?
5. What is your greatest strength?
6. What are your strengths and weaknesses?
7. What do you know about this company/organization?
8. Why should we hire you?
9. What is your greatest accomplishment?
10. What are your salary requirements?
11. Do you have any questions for us?
12. What are you looking for from a new position?
13. Are you considering other positions in other companies?
14. What is the professional achievement you're most proud of?
15. What kind of working environment do you work best in?
16. Where do you see yourself in 5 years?

## Situational Job Interview Questions:

1. Why haven't you gotten your Bachelor's Degree/Master's Degree/Ph.D.?
2. Why have you switched jobs so many times?

[Did you change your career path?](#)[Register](#)[Sign In](#)[Why did you decide to leave your previous/current job?](#)

5. Why is there a gap in your work experience?
6. Why were you fired?
7. How do you feel about working weekends or late hours?
8. How would your boss describe you?
9. Do you have any serious medical conditions?
10. What would your first 30, 60, or 90 days look like in this role?
11. Are you a team player?
12. Are you a risk-taker?
13. How do you deal with pressure or stressful situations?
14. Do you think there is a difference between hard work and smart work?
15. How quickly do you adapt to new technology?
16. Do you have any interests outside of work?
17. What do you think our company/organization could do better?

## Behavioral Questions in a Job Interview:

1. Give an example of how you have handled a challenge in the workplace before.
2. Give an example of when you performed well under pressure.
3. Give an example of when you showed leadership qualities.

## How to Answer 14 Most Common Job Interview Questions [+ Sample Answers]

These questions are the ones you're bound to hear at [just about any job interview](#), whether you're an intern or a senior professional with a decade of work experience.

All of these questions are used to learn more about you, both as a person and a professional.

You might have heard the popular idea that there's no right or [wrong answers for interview questions](#).

Well, while that might be true, there ARE a set of rules you need to follow when answering these questions.

# Jobseeker's Dream Duo:



[Get Your Copies Now](#)

If you understand what, exactly, the interviewer is looking for with each question you'll be able to give the right answer (and rock that interview!)

In this section, we're going to go through 14 of the most common job interview questions and answers. We're going to explain what the HR manager wants to see you, as well as give you sample answers you could use.

So, let's get started!

## 1) Tell me something about yourself.

How hard can it be to talk about yourself? We do it on a daily basis without much thought to it.

However, recruitment managers are not looking for your whole life story, your third grade achievements, or what you had for dinner last night. Instead, **they are looking for a pitch.**

This is usually the first question asked in an interview, so it acts as your introduction. Make sure your answer is **relevant to the position** you are applying for. What you should be aiming for here is to **present yourself** as the ideal candidate for the job.

A good rule of thumb is to structure your talking points as follows:

- **Briefly introduce yourself:** What's your name? How long have you been working in [profession]?
- **What do you love about your job?**



` are your top 2-3 achievements that

[Register](#)[Sign In](#)

apply

Now, let's go through some examples:

### Possible Answers for "Tell me About Yourself":

#### Sample Answer 1:

*Hey! So, my name is John Doe and I've worked as a business analyst for 5+ years in Company X and Company Y.*

*I have some background in data analysis, having studied Information System at [Made-Up] University.*

*Throughout my career, I've done some pretty impressive stuff (if I do say so myself, haha).*

*For example, at Company X, I led a project for migrating all operations data to a new data warehousing system to cut down on costs. The new solution was a much better fit for our business, which eventually led to savings of up to \$200,000 annually.*

#### Sample Answer 2:

*I am Jane Doe, a recent college graduate from the University of Wisconsin-Madison.*

*I have just graduated with honors in Biochemistry. I know my way around a lab and have had multiple opportunities to put my knowledge into practice as a chemistry research assistant.*

*The lab felt like home, which is why I'd love to work as a lab assistant. I am passionate, hard-working, and extremely responsible. I am also looking forward to putting to practice all the things I learned during my time at university.*

## 2) How did you hear about this position?

Although at first glance this might seem like a straightforward question, you should grab any opportunity you can to **show your interest** in the company.

Even if you haven't been continuously refreshing the company's website for job listings, make it seem like you have (in a professional way, of course). Show **excitement** and curiosity.

If someone inside the company told you about the position or **recommended** the company, definitely make sure to mention that.

You'll have a much better chance at getting hired if someone credible can vouch for your skills.

[ion his/her name and his/her posit](#)[Register](#)[Sign In](#)

their reasoning for inviting or recommending you to apply for the position. Tell the hiring managers what excites you about the job opportunity or what exactly caught your eye.

### Possible answers for "How did you hear about this position?":

#### Sample answer 1:

*"I've known about [MadeUpTechnologies] for a long time - I'm a big fan of your products. I even own one of your latest phone models!"*

*I love the company's passion for creating super intuitive, beautiful hardware, and I would love to be a part of it.*

*So, when I saw your job ad at [RandomJobBoardWebsite], even though I wasn't actively looking for a job at the time, I couldn't help but apply!"*

#### Sample answer 2:

*"I heard from Jim Doe, my old colleague and college friend, that [Company X] was looking for a new sales director. He encouraged me to apply, saying that my experience managing a sales team at [Some Software Company] would be helpful for [Company X]."*

*I've heard a lot about [Company X] from Jim, and I'm a big fan of the way you do things there. I've always wanted to work for a company with a flat organizational structure."*

## 3) Why did you decide to apply for this position?

Through this question, the interviewers want to assess how passionate you are about the position. And no, the answer isn't:

*"Well, I'm very passionate about not starving to death."*

Or...

*"Well, I needed the money, and you guys tend to pay a lot."*

What the interviewer is looking for here is to see how passionate you are about the job or the company. After all, job performance is directly linked to job satisfaction. The happier you are about your position at the company, the more productive you will be.

And here's the kicker - your passion will be very evident during the interview.

When you're talking to a person that's passionate about something, you can pretty much feel them glow as they talk. And if you're an HR manager who's interviewed hundreds of people, this is a very good sign to hire the candidate.



his knowledge to your advantage.

[Register](#)[Sign In](#)

When asked this question, your answer should include 2 things:

1. What motivated you to apply for this position, **specifically**.
2. Why this company? Have you heard of them before?

### Sample Answer 1:

*I'm very passionate about sustainability and renewable energy. In fact, I minored in Environmental Science at [XYZ University].*

*I've always wanted to put my engineering degree to a good cause - and the position as a Sustainability Coordinator at [Company XYZ] is just the right thing.*

*I've been following your company for the past few years, and I love how you're changing the renewable energy landscape in America.*

Keep in mind, though, that if you don't know much about the company or the position - that's OK too. Just be honest and show your passion for the job. However, it's always better to do your homework before going to an interview..

### Possible answer 2:

*I've always wanted to get into marketing. Having done promotional jobs here and there, I never had an opportunity to do something more serious.*

*I do believe, though, that I have just the right skills to get started: copywriting, basic photoshop, and of course, lots of creativity.*

*So, I thought that [an internship](#) at [Company X] would be an awesome start to my career in marketing.*

*Want to find more samples answers to this question? Check out our article on 10+ answers to "Why do you want to work here?"*

## 4) What are your biggest strengths?

There are two answers you could go for here: what your actual strengths are, and what you think the hiring manager or HR representative wants to hear. We would most certainly suggest you go with the first answer.

For this question, you would want to narrow your answer down to **at most three strengths**. Pick 1 or 2 skills that would help you really excel at the job, and 1 or 2 personal (more or less unrelated) skills.



which ones are your top strengths?  
one's perfect for your field:

[Register](#)[Sign In](#)[Learn More](#)

<b>Analytical Jobs</b> Analyst, Software Dev, etc.	Critical Thinking	Analytical Thinking	Problem-Solving	Detail-oriented	Logical	
<b>Creative Jobs</b> Designer, Marketer, Advertiser, etc.	Creative	Originality	Open-Mindedness	Detail-oriented	Curiosity	
<b>Management Jobs</b> Pretty much anything related to managing a team	Leadership	Organization	Communication Skills	Persuasion	Teamwork	Detail-Oriented
<b>Hands-On Jobs</b> Waiter, Cashier, etc.	Hard-working	Detail-Oriented	Motivated	Multi-tasking	Problem-Solving	Punctual
<b>Communication Jobs</b> Sales, Customer Support, etc.	Charisma	Communication Skills	Energetic	Persuasion	Witty	Social

After picking your strengths, **back it up with a situation** or story that shows how you have used it to benefit you on the job.

After all, words are just that - words. The HR can't know whether your "natural leadership" is an actual strength, or just means that you were super active in your high school class.

As you probably already know, this is one of the most common interview questions out there, so make sure you're prepared for it before facing the HR manager!

### Possible answers:

#### Sample Answer 1:

*My biggest strength is that I'm good at picking up new skills. I've worked a variety of different odd jobs - things like working as a waiter, house-keeper, cook, and a lot more (as you've probably seen on my resume).*

*For most of those jobs, I ended up picking up all the needed skills within 1 or 2 weeks (with basically no previous experience).*

*So, I'm pretty sure while I don't have any experience as a bartender, I have the right certification, and I believe I can get good at it within a week or two.*

#### Possible answer 2:

*My biggest strength is that I'm very efficient at working under pressure. No matter the crisis or stress, I can make the right decisions on-the-spot.*

*As an event manager at Company X, we were organizing an IT conference for a client. There were a ton of last-minute hiccups - some speakers canceled and the catering company said they'd be late for the lunch break. On top of that, we were understaffed because 2 of our volunteer organizers got sick and couldn't show up.*

[Register](#)[Sign In](#)[.he](#)

at point, things looked so bleak that I considered quitting or leaving the event or postponing it. Instead, I took the initiative in my hands and sorted through the problems one by one.

## □ 5) What is your biggest weakness?

Ah, this is always a tricky one!

After all, you don't want to mention your flaws during an interview, so it's guaranteed to be a tough question.

The trick to answering this one is realizing that the interviewers don't expect you to be perfect. Everyone has flaws, weaknesses, and things to improve on.

When asking this question, the HR manager is actually seeking to learn:

- Whether you have the right skills for the job. If you're applying for the position of server in a busy restaurant, and you say your biggest weakness is performing under pressure, then you're definitely not getting a callback.
- If you're self-aware and really know what your sticking points are.

And NO: **fake humble-brag** weaknesses don't count as weaknesses. You can't just say that your biggest weakness is that you work too hard, or that you're a perfectionist.

The key here is to **mention a weakness that's real**, but not something that would get in the way of you doing your job. You wouldn't want to say you're bad at math if you're applying for an accountant position, would you?

It's also good practice to mention how **you are working towards** overcoming this weakness and realizing how it affects you negatively. If you can, **just balance it with a positive side effect**: treat it like two sides of the same coin.

### Possible answers:

Sample answer 1:

*My biggest weakness has always been my communication skills. I've been pretty shy and anxious as a kid. Over the years, however, I've been really working on the issue.*

*At this stage, I'm much better than I've ever been, but I'm still far from perfect.*

*This, however, won't have any impact on my job as a programmer. Despite lacking communication skills, I'm very good at working in a team.*

Sample answer 2:

*Well, as a recent graduate, I'd say my biggest weakness is the lack of real-life*



experience.

Register

Sign In

While I've worked on a dozen software projects in the university, I don't have the experience of working in a fully agile environment with an experienced team.

I am, however, willing to do my best and catch up as fast as I can.

Looking for more samples answers about your [strengths and weaknesses](#)? Check our full guide!

## 6) What do you know about this company/organization?

A quick search in the “About” page of the company/organization you are applying for should be enough, right? Well, yes and no.

Think of this as an open-ended question. There's no real wrong answer here, other than:

### Incorrect Example

*I don't know anything about this organization. In fact, how did I end up here? Can you guys call me a cab real quick?*

**However**, the more you actually know about the company, the better your chance of getting hired.

Imagine 2 equally competent candidates:

1. One who doesn't particularly care much about your company, and is only applying because they know you pay good salaries
2. Another who's been following your company blog for ages, loves your product, has several friends already working in the company

Which one would you pick? Exactly, the second one!

So, with this job interview question, you want to convince the recruiter that you're candidate #2.

Now, how do you do that? Well, a rule of thumb here is to do some Googling before the interview and learn the following about the company:

- What does their product or service do?
- What impact does the product / service have?
- What's the company culture like?
- What are the latest news about the company? How are they performing?



## Possible answers:

### Sample Answer 1:

I hadn't heard about you until recently, actually. I found out about [Company] through your job ad on RandomJobBoard.

After doing some brief research on you guys, I ended up falling in love with yo software and your mission.

Now, I've worked with a ton of different project management software - Example Software 1, Example Software 2 - but none of them were as intuitive and as Example Software 3.

### Sample Answer 2:

Well, I know that you're one of the biggest investment banks in [town / state / country]. Company X pops up on news pretty often - I've read that you've invested in some of the hottest tech IPOs, and have several up-and-coming biotech companies in your portfolio.

I got particularly interested by your recent investment in [Startup X], I found that interesting because of [Y Reason].

## □ 7) Why should we hire you?

Ah, the ultimate humble-brag question.

Now, the real question is, how do you sell yourself without trying to look arrogant desperate, or needy?

A good rule of thumb here is to stay away from the extremes. Think you're a good for the job? Say that "you have the right experience."

Whatever you do, don't oversell yourself:

*"I'm the best salesman you've ever met!"*

Instead, make a general statement (I'm a great fit for the position because...) and about your experiences and achievements.

Here are 3 general points you can mention:

1. How you're super passionate about working for the company (and why).
2. How your skills fit their requirements.
3. How you're going to help the company solve their existing problems. Improve a metric, setup a process, etc.



Register

Sign In

## Possible answers:

### Sample Answer 1:

*Well, as a start, I have all the skills and work experience required for the job. I've worked as a Sales Manager for 5+ years, and over the past 2, I've closed several deals totalling in 6-figures.*

*Oh, and on top of that, I have experience working with tech companies, so I'll be able to pick up all the product specifics much faster than the other candidates.*

### Sample Answer 2:

*I have just the right skill-set to excel as an executive assistant. While I haven't previously worked as a personal assistant, I pretty much fit the bill for the role*

*I'm extremely organized, having managed several project teams in my university. I led the organization of Event #1 and Event #2. This involved continuous communication with 12+ companies, 30 speakers, and 15+ sponsor*

*I'm very meticulous and organized, and I'm more than capable of helping the CEO get the most out of their free time.*

Looking for more sample answers? Check out these 10+ answers to “[Why should I hire you?](#)”

## 8) What are your salary requirements?

This is always a tricky question. You don't want to lowball yourself, but at the same time, you don't want to be told “No” because you gave such an outrageous num

When answering, keep these 3 things in mind:

- What's the average salary for someone of your skill-level?
- How much does the company pay employees of your skill level? [GlassDoor](#) should be super helpful here.
- Finally, how much are you getting paid in your current company? In most cases you can probably negotiate a pay bump from what you're currently getting.

The final number you tell them should incorporate all 3 of the points we just mentioned. Do you know for a fact that the company is doing well (and compenses employees accordingly)? You'd quote a higher salary.

Is your skill-level above average? This should be reflected in your salary.

As a rule of thumb, you can figure out 2 numbers: what's the “good” scenario, and what's the “best” scenario?

Answer the interviewer with your “best” pay, and worst case scenario, they'll negotiate it down.



can also answer with a range, and c somewhere in the middle.

[Register](#)[Sign In](#)

er

### Possible answers:

#### Sample Answer 1:

*My salary expectation is around \$70,000 annually.*

#### Sample Answer 2:

*My salary requirement is in the \$30,000 - \$40,000 range annually.*

## 9) Do you have any questions for us?

You'll hear this question in **every interview** you will attend.

While there isn't a right answer, there IS a wrong answer:

*Nope, all good! Thanks, I'll go show myself out.*

Instead, with this question, you want to show your enthusiasm about the company. Imagine they've already hired you and you're starting tomorrow - what would you like to know about them?

Keep in mind, though, that the questions shouldn't be too easy (So, what does your company do?).

Other than showing the recruiter that you're really interested in working for them, this is your opportunity to really find out more about the ins and outs of the place.

The answers you get from the interviewer could also be an indicator of **whether they really want to work there or not**.

So, what kind of questions can you ask? Here are some of the most essential ones:

### Possible questions to ask at the end of an interview:

1. What does a regular day in this company look like?
2. What's the best thing about working for the company?
3. What's the worst thing about working for the company?
4. What would you say are the biggest challenges a person in this position



t face?

Register

Sign In

What are the most important skills and qualities a person in this position needs to have?

6. What do you like best about working in this company?
7. What are the most pressing issues and projects that need to be addressed?
8. Do you have training programs available to employees?
9. What sort of budget is there for my department?
10. What kind of opportunities do you have for future development?
11. What are the performance expectations for someone in this position?
12. Do departments usually collaborate with one another?
13. Do you celebrate birthdays or retirements in the office?
14. Do employees usually hang out with each other outside of work?
15. Is there anything else I can help you with at this stage?
16. What is the next step in the hiring process?

For the complete list of [all the questions you can ask the interviewer](#), check out our article!

## 10) What are you looking for in a new position?

The easiest way to answer this question would be to simply say that you're looking for whatever the company is offering.

Look at it from the point of view of the potential employer. Would they hire someone if they answered this question with:

*A good salary. And uhh, well, that's about it!*

This answer pretty much says that the moment they get a higher paying offer, they're going to jump ship!

Instead, explain to the interviewer that **this job at this company** is the perfect fit for you. Mention what your short-term and long-term [career goals](#) are, and how this position ties to them.

### Possible answer:

*I'm looking to further apply my machine learning skills that I developed during my 2+ years of work at [Startup X]. There, I used to do programmatic ads modeling and design.*

*Now, I'm looking for an opportunity to work on a larger scale project that involves setting up programmatic ads for audiences of more than 10 million people.*

*I believe that worked with such a large-scale project will allow me to progress*

*I significantly faster in my career.*[Register](#)[Sign In](#)

## 11) Are you considering other positions in other companies?

Here's a tricky one: How much does the HR manager need to know here?

If you admit to having interviews with other companies, it might look like you're not 100% dedicated to this one.

On the other hand, if you say you are not considering other positions, it might make you seem like you don't have other options (and the company has the upper hand in salary negotiations).

The right way to go about here is to find common ground between the two answers.

The interviewer is probably asking because they want to know whether they have competition in hiring you. They also want to know if you are serious about the interview and are legitimately looking to be employed in this field of work.

If you do have other interviews lined up for other companies, express that you are **keeping your options open** but that you favor this job in comparison to the other ones.

Don't have many other options? Stick to the same approach.

Whatever you do, don't make it seem like you're desperate or that you don't have other options.

### Possible answers:

#### Sample Answer 1:

*I have had two interviews during the past week with companies in X and Y industries.*

*However, as I'm very passionate about both your industry and the work you have done during the past several years, I am more inclined toward working for you, if everything works out.*

#### Sample Answer 2:

*Not yet - I wasn't really actively looking for a job until my friend, [name], recommended your company. I'm not looking for just any company - I'm interested in an interesting, engaging project such as yours.*

## 12) What is the professional achievement you're most proud of?

This is another version of the "Why should we hire you" question, but with a focus



→ specific achievement.

[Register](#)[Sign In](#)

This one's pretty straightforward, just mention your #1 professional achievement you're good to go.

As a given, the achievement has to be related to the job you're applying for. Let's say you're applying for the position of **Sales Manager**:

- **[Incorrect Example]**

*"I'm very good at underwater basket-weaving, having woven 20+ baskets in the past year."*

- **[Correct Example]**

*"In my previous sales position, I managed to hit and exceed department KPIs by 50%+ for 6 months in a row"*

Keep in mind, though, that you want to be very specific with your answer. To get it right, try using the **STAR method**. It goes something like this:

**S: Situation** - Set the scene and context.

**T: Task** - Describe what your challenge or responsibility was.

**A: Action** - List and dwell on all the actions you took towards addressing the challenge or responsibility.

**R: Result** - Explain what the outcomes were and how they fit with the overall goal of the project or company.

So, find a work-related achievement that showcases your contribution through your skills and experience to something that matters to the company.

## Possible answers:

### Sample Answer 1:

*My biggest achievement is the fact that I went from being an intern to managing company X's entire marketing over 2 years.*

*As an intern, I basically had 0 instructions on what to do - it went like "hey, go learn social media advertising and get it going." The founders didn't exactly expect me to achieve much, and didn't particularly care, as they were 100% focused on making the product work.*

*Instead of just complaining about a lack of direction, I started reading up on digital marketing - pretty much anything I could get my hands on. I learned*

*how to do content marketing for example, from Neil Patel's blog, and started putting everything into practice.*

*My first success was getting an article to go viral, generating over \$5,000 revenue in a single day. While that's not much for a software company, it felt*



... a lot for an intern.

[Register](#)[Sign In](#)

After that, the founding team gave me a lot more trust, and assigned me a small marketing budget of \$1,000 per month. With a lot more confidence in my abilities, I started experimenting with other strategies.

Then, over the next 2 years, I got promoted to Head of Marketing. After making a couple of hires, I managed to scale up our marketing efforts, growing the company from \$2,000 to \$30,000 monthly recurring revenue.

#### Sample Answer 2:

My greatest accomplishment so far is graduating from [University X] within 4 years, with a GPA of 3.9. My family was unable to support me financially, so I had to take care of all the university bills on my own.

Through hard work and dedication, I ended up graduating with almost no student loans. I managed this through a combination of:

1. Working part-time while studying
2. Doing seasonal full-time work during the summer
3. Maintaining a high CGPA and winning 2 scholarships over 4 years

## 13) What kind of work environment do you like best?

The aim of this question is to assess whether you'll **fit in the company's working environment**.

For example, some organizations are pretty structured and hierarchical, they require tight organization and have a well-planned day filled with rules and guidelines on how to do things.

If you're the creative, think-out-of-the-box type who likes to break the rules and innovate, this is probably not going to cut it for you.

On the other hand, some companies are more laid back, with a lot less bureaucracy. "Go get us more sales" can actually be your main duty for the week if you're working in an early stage startup.

If you're the type who prefers to have strict to-dos and objectives, you probably enjoy such a job.

So, the takeaway? Different people work best in different environments, and that's okay.

Before you go to the interview, go through the company's website and **social media** pages to get a sense of the general **vibe** and **environment** there.

Look at **employee** reviews on GlassDoor, or if you know someone already working there, ask them.



## Possible answers:

### Sample Answer 1:

*I work best in smaller companies. I really dislike the corporate world - rules, guidelines, SOPs, and so on. I perform best when I have a certain level of freedom to do things. Want to find innovative solutions to problems you didn't even know you had? I'm your guy.*

*Want someone to just blindly follow instructions and do what they're told? Then we'd probably not be a good fit.*

### Sample Answer 2:

*I love working in a youthful, energetic environment. You know, when you're working on a common goal with a team of people who are as passionate as you are?*

*I like to think of my work as a second home, and my coworkers as family.*

*The last company I worked at had such an environment, and I excelled at the job.*

*I get that exact feeling about Company X, since the moment I walked in here for the interview. So, I'm pretty excited to get to know how you guys work!*

## 14) Where do you see yourself in five years?

Sometimes the honest answer to this is “*Hopefully not doing this.*” especially with entry-level jobs.

Don’t think the hiring manager doesn’t know it, though. There are diplomatic ways to go around it.

In general, the motivation behind this question is for the interviewer to assess whether you are **an ambitious person** or not and whether you have **realistic expectations** for your career.

Make sure to avoid any of the cliche answers such as...

“*In your seat!*”

Or

“*As the big boss man*”

Instead, think realistically about what **the next step** after this position is, and whether it is possible to **reach it within the company** you are applying at.

**possible answers:**[Register](#)[Sign In](#)**Sample Answer 1:**

*Within the next 5 years, I'd like to reach the position of a Senior Business Consultant. During the time period, I would like to accomplish the following:*

*Help 20+ organizations improve their business*

*Create a personal network of highly specialized professionals*

*Learn as much as I can about optimizing and improving clients' businesses, as well as the essentials of operating a company*

**Sample Answer 2:**

*As a start, I want to learn if accounting is the right field for me. While I loved what I studied at the university, I want to see if working in the field feels the same.*

*If I do end up enjoying it, I'd like to specialize in either internal auditing or forensic accounting, as I really like to discover and solve problems. From what I've seen from your job ads, you guys are hiring for both, so I hope it's going to be possible to move up from the position of an "intern" within the next few months!*

*Still not sure how to answer this one? We don't blame you! Sometimes, you might know what you're doing next week, let alone next year! Check out our guide to answering the "[Where do you see yourself in 5 years?](#)" job interview question to 1 more possible answers.*

## **How to Answer 18 Situational Job Interview Questions**

Congrats!

You're past the hard part.

You already know the most common job interview questions, and can probably deflect whatever the interviewer throws at you.

Depending on your specific situation, though, you might also need to learn how to answer these situational job interview questions...

### **1) Why haven't you gotten your Master's Degree/Ph.D.?**

As a start, keep in mind that the interviewer isn't judging you for your decision.



if they were looking for someone w [Register](#) [Sign In](#) ha  
invited you to an interview. The degree is not the dealbreaker here, but your ans  
to the question might be.

When asking this question, the interviewer is trying to see your reasoning for purs  
a career instead of getting another degree.

Heck, there's a chance that if you give them the right answer, they're even going  
like you more than someone with 3 Phds!

So, simply explain why you didn't think that another degree was the right thing fo  
you at the time.

Don't say you were lazy or didn't feel like it, or that it's a waste of money (even if i  
might be the case).

Instead, give compelling arguments, such as...

- You wanted to see whether your field was the right one for you.
- You didn't have the financial resources at the time.
- You wanted to get some practical work experience before committing to ano  
degree.

## Possible answers:

### Sample Answer 1:

*At this stage of my life, I decided to pursue my career instead of further  
education. On the one hand, I want to make sure that Marketing is what I wan  
to do with my life.*

*On the other hand, I believe that in my field, practical work experience is a lot  
more valuable than academic.*

*So far, my decision has paid off pretty well - I've already gotten a lot of  
experience doing online marketing for 3+ companies and delivering awesome  
results to boot.*

*I might eventually decide to pursue a masters, but at this point, I really don't  
see the point in that.*

### Sample Answer 2:

*Because it's not in sync with my future career path. I believe that for software  
engineering, practical experience matters a lot more than having a degree.*

*I've already done an internship as a Junior Javascript Back-end Developer, an  
I believe that it gave me a lot more knowledge than my B.A. in Computer  
Science.*

*While I am eventually planning on getting a Masters, it's going to be in a more  
theoretical field, such as Artificial Intelligence.*



Register

Sign In

## 2) Why have you switched jobs so many times?

If you've switched jobs in a very short period of time (2 or more full-time positions per year), the interviewer is bound to ask about it.

After all, job-hopping is one of the biggest red flags for HR managers.

True, you might have had a reasonable cause. Maybe the second company you were hired in just wasn't a good culture fit for you.

Well, you'll have to communicate that.

Companies tend to be skeptical because of the following reasons...

- You might be a **job hopper**. Some people tend to switch jobs the moment they get a better salary offer.
- You might be **unqualified** for the job and you quit because you couldn't deliver.
- You get **bored easily** and your solution to that is quitting.

So, your job here is to convince the interviewer that you don't belong to any of these 3 categories.

You need to make them realize that **you will not jump ship** a few months after getting hired just because some recruiter PM'd you on LinkedIn with a better offer.

The best way to answer this question is to explain the reason you switched jobs. It could be one of the following:

- The company culture wasn't a good fit. This happens to the best of us - sometimes the company just isn't the right one.
- The job description was misleading and you ended up doing something you either didn't enjoy, or were not qualified for.
- You learned that you simply didn't enjoy the job, and are not willing to try out something different. While this isn't the best potential answer, it's honest and chances are, the HR manager will understand.

### Possible answers:

#### Sample Answer 1:

*The last company I got hired in just wasn't what I expected. The hiring manager didn't communicate the role well enough.*

*As you already know, I'm a copywriter - I write sales copy. I work with:*

*-Landing pages*

*-Email marketing*

*-And sales pages*

[Register](#)[Sign In](#)

I'd a week after I started work at the company, I realized I was actually looking for something completely different. They asked me to write generic blog and social media posts, which is pretty far off from what I do.

This was really not what I expected, and not something I find interesting.

#### Sample Answer 2:

Well, as a start, my first job was in a big corporation straight out of university. While I did learn a lot there about Software Engineering practices, I also learned that a huge company with lots of regulations, rules, and the like isn't for me.

So, at the end of my internship there, I decided to try working at a startup. I enjoyed that job a LOT more, as it gave me a lot of freedom when it comes to problem-solving. I wasn't told HOW to do it. Rather, I was given the option of coming up with my own solution.

Unfortunately, the company went belly-up after failing to raise money, putting me back on the job market.

And here we are - [Company X] is pretty much THE place I've always wanted to work in. I've heard a lot about your company culture, and thought I'd really belong there.

## □ 3) Why did you change your career path?

If you recently [changed your career path](#), the interviewer is sure to ask about it.

Don't worry - there's nothing wrong with this.

A lot of people go through a [career change](#). Some even do it several times in their lifetime!

As long as you're good at what you do, no one cares if you were a pediatrician in one year, and a professional chef in another.

When asked this question, all you have to do is answer truthfully. Explain how your job just wasn't for you, and how the job you're applying for is so much more interesting.

#### Possible answers:

##### Sample Answer

I realized that being a doctor is not for me. While I did enjoy my 3 years in medical school, the 6 year study period was too much.

I wanted to start making money and help out my family way before that, so I dropped out of university and started taking online courses in accounting.



*...s point, I'm pretty good at it, having worked at Firm X and [Company Y].*

[Register](#)[Sign In](#)[Job Search](#)

### Sample Answer 2

*Simply because I enjoy doing sales much more than accounting. After 5 years of working as an accountant for Firm X, I decided I wanted to try something new.*

*I asked my boss at the time to let me transition to the sales team, and I ended up liking it AND being pretty good at it.*

## 4) Why did you decide to leave your previous/current job?

When asking this question, the interviewer wants to learn:

- **Did you have a good reason for leaving your last job? The HR manager doesn't want someone that just jumps ship the moment things go bad.**

### Incorrect Example:

*"Oh, well, the company started bleeding cash and was on its way to bankruptcy."*

### Correct Example:

*"I felt like it was time - I got to a point where everything I was doing felt monotonous. I learned as much as I could at this position while delivering amazing results. It was, however, time to switch to something new."*

- Did you leave on good terms? Meaning, did you go through the offboarding process, instructing your coworkers on how to take up your responsibilities? Or did you just say "Adios" and stopped showing up at work?

### Incorrect Example:

*"Things started to get really boring, and the boss man was kind of mean. I totally deserve better, so I just ghosted them and now I'm looking for a new company. Hi!"*



Register

Sign In

### Correct Example:

*"I didn't feel like the company's values coincide with mine. The management was too controlling and micromanaging. I prefer to have some control over my work, and being able to contribute by going above and beyond my requirements."*

Of course, I went through the off-boarding properly. Meaning, gave a [timely resignation notice](#), and transferred all the essential company knowledge to my replacement.”

- Did you leave voluntarily, or were you fired?

### Incorrect Example:

*"I got fired for missing work for a week without an excuse."*

### Correct Example:

*"I was fired, actually. The fault was in my communication skills at the time. I misunderstood my supervisor's instructions and ended up setting a higher monthly spend on ad account for the client. The losses were not more than 3-figures, but apparently, the relationship with the client was already strained, so they ended up leaving."*

*Of course, I really took this to heart and worked very hard on improving my communication skills, to ensure that I don't make any mistakes of this nature ever again."*

## 5) Why is there a gap in your work experience?

In most cases, a gap in your work experience doesn't really mean anything. You probably have a very good reason for it.

The interviewer, however, will definitely ask about it, and you should answer adequately.

There's no secret sauce to answering this question, just let the recruiter know about your situation, whatever that may be:

- Maternity leave
- Health issues



g for a sick family member  
off to pursue further education

[Register](#)[Sign In](#)

- Relocating to a different city
- Working on a personal project

Whichever the case may be, just explain the situation in brief and move on.

One thing to keep in mind, however, is that if you were laid off at work, or you quit and had trouble [getting a new job](#), you should be very subtle about it.

If the interviewer knows that you're struggling to [find a job](#), you're going to give them the upper hand in salary negotiations.

### Possible answers:

#### Sample Answer 1:

*"I had a baby and had to take maternity leave."*

#### Sample Answer 2:

*"My father was sick, so I had to be the one to take care of him full-time over a few months."*

## 6) Why were you fired?

Now this is a tough one.

Getting fired is pretty much never good.

Keep in mind that we're talking about getting fired, not getting laid off. There's a huge difference between the two:

- **Getting laid off** means that you got let go for something that had nothing to do with your competence. *I.e. budget cuts, company down-sizing, etc.*
- **Getting fired**, on the other hand, means that you got let go for a reasonable cause. And chances are, it's probably your fault.

If you got fired and the interviewer asks you about it, you should be honest. After all, they can easily check-in with your previous employer.

Your best shot here is to be critical about your mistakes, and explain what you've done to improve.

### Incorrect example:

*"None of it was my fault. My boss is a total tool, and he hates me for no real reason. He yelled at me for no real reason!"*



example above, the interviewee g  
... for the HR manager.

[Register](#)[Sign In](#)

re

*Instead, try saying something that shows that you're aware of your mistakes."*

### Correct example:

*"The main fault was in miscommunication. The interviewer was unclear about the job responsibilities - from what I understood, they were looking for a senior level marketer to oversee their email marketing operations.*

*At the end of the day, though, it turned out that the company was looking to experiment with email marketing, and specifically for someone to set it up from scratch.*

*While I did my best to deliver, in the end it turned out that their niche doesn't actually need email marketing. This was against the management's vision, so that decided to let me go."*

## 7) How do you feel about working weekends or late hours?

You're gonna get asked this question in one of the following 2 cases:

### 1) You're applying for a job that requires working odd hours.

In this case, your answer is pretty straightforward - since you're applying for such a job, you probably don't have any problems working odd hours.

### Sample answer:

*"Sure! I'm OK with working late hours or weekends, as long as you let me know about it at least a few days in advance."*

### 2) You're applying for just about any other type of job.

Now, you should look at this as a red flag. Is the employer just checking your dedication, or are they looking for someone that's going to work 24/7 with no overtime pay?

In this case, ask them to clarify what they mean.

### Sample answer:



n enough warning, sure. Is that som... Register

Sign In

? D

you offer overtime pay for this kind of situation?"

## 8) How would your boss or coworkers describe you?

This question is pretty much the same as “[what are your greatest strengths](#),” the difference is that it should be from the point of view from your boss or coworkers.

Here, you want to focus on your traits and achievements that you’ve previously been praised for (After all, the interviewer might ask for a reference!).

There are at least 2 ways to answer this question:

**1) Describe a specific situation where you excelled at work (and received praise from your boss and coworkers)**

### Sample Answer

*“They’d say I’m super hard working. During my weekend-off, not one, but three of my coworkers got sick, and I had to spot for them.*

*The weekend was peak season in Nantucket, so the restaurant was getting seriously overwhelmed. All of a sudden, we went from being very prepared for the season, to complete panic.*

*Had to jump between serving, bussing, and line-cooking, but overall, managed to survive through the weekend successfully.”*

**2) Quote a performance review**

If you’ve previously worked in an office job, you’re probably all too familiar with this:

Did your boss give you a glowing performance review? Make sure to mention it here.

### Sample answer:

*“Well, in my last performance review in September, my boss described me as someone who takes initiative.*

*My position as a PR manager involves constantly keeping track of our clients’ brand reputation, and if something goes wrong, dealing with it as fast as possible.*

*In a lot of cases, you need to be very proactive - if you wait for your entire team to notice a problem, it’s already too late.*

[Leave a meeting on how to deal with th...ate.](#)[Register](#)[Sign In](#)

*There were 4-5 different situations where I had to take charge and react to problems literally the moment they arose, whether it was during my work hours or not."*

## 9) Do you have any serious medical conditions?

When asking this question, the interviewer wants to learn if you have any medical conditions that could impair your ability to do the job correctly.

In most cases, you're not obliged to give an answer. If you do have a health condition and it doesn't have anything to do with your career, you can simply choose not to answer, or to say "No."

However, you might want to disclose anything that could potentially have an impact on how you perform.

For example, if the job requires you to lift heavy boxes, for example, and you're unable to do so because of a condition, you should let the HR manager know.

### Possible answers:

#### [Sample Answer 1]

*"I don't have any serious medical conditions"*

#### [Sample answer 2]

*"I'm unable to lift heavy objects because of issues with my back, but it won't have any impact on how I perform at an office job"*

## 10) What would your first 30, 60, or 90 days look like in this role?

If you're applying for a senior or leadership role, you're probably going to get asked this question.

Chances are, at this stage of the interview, you already know a lot about your future position and the company.

Now, it's time to show off your knowledge in your field, and explain how you're going to start making things happen at the company.

So, here's how to answer the question:

**For the first 30 days:**[Register](#)[Sign In](#)

You're probably going to need to get to know the company first. You're going to be learning as much as possible, including information on:

- What does the company do?
- What are the key processes?
- What does your department do?
- What are the current problems and challenges?
- Where can you help?

### **Then, during the 60 days:**

You'll start making things happen. From all the info you gathered, suggest a handful (3 to 5) initiatives you could take on:

- You'd audit the company email marketing strategy and suggest improvements
- You'll help come up with better ad copies for Facebook marketing
- You'll help the team with their ongoing marketing initiatives

### **Within the first 90 days:**

You're already have started making an impact. Describe several things you think are going to be functioning better:

- Online ads are going to be performing better by 10-20%
- Email marketing operations are going to be more streamlined, taking significantly less manpower

## **□ 11) Are you a team player?**

Wherever you're applying, the answer to this question should be a “Yes!”

Even if you're applying for a completely solo role, chances are, you're still going to have to work in a team **occasionally**.

We'd recommend being very specific about your answer here - don't just say yes. Give the interviewer an exact example of when you excelled at working with a team.

#### **Possible answers...**



Register

Sign In

**Sample Answer 1:**

*"I'm much better at working in a team than alone, actually. That's what I love about working in advertising - everyone has their own specific type of a creative spark, and when you combine it all, magic happens!"*

*"I'm good at both leading and following in terms of creativity and brainstorming. I'm also super receptive to others ideas, and do my best to help them execute it without nay-saying or criticism."*

**Sample Answer 2:**

*"Yep, definitely. I excel at team-work."*

*This one time while working at [Company X], I was assigned to an existing team working on a web application for a business process management company.*

*They were working on a tight deadline, and needed help on the API side.*

*I optimised their development cycles and oversaw a team of three developers while collaborating with the other two dev teams.*

*Everything went pretty well, and we managed to finish the project on time."*

## 12) Are you a risk-taker?

This one's pretty tricky, as the answer here depends on your profession and field.

Ask yourself - is risk-taking a valuable skill for the job?

If you're a pilot, for example, the answer should be a strict "No!"

If, on the other hand, you're a day trader, then risk is an essential part of your job

So, depending on how valuable risk is for your job, answer accordingly.

You could also give a more strategic answer. Let's say, for example, you work in investment banking. You need to be a risk taker to an extent, but being too risk-friendly might make the entire company go bankrupt.

The strategy in such a case would be to show that you're all about **calculated risk**. You're willing to take chances, but only when the odds are in your favor.

As with most interview questions, you should give examples of situations where you had to take risks, and what the end-results were.

### Possible answers:

**Sample Answer 1:**



I'm a risk-taker. I believe that to act  
I'm willing to take a certain level of risk.

[Register](#)[Sign In](#)[d to](#)

Pretty much any marketing initiative you launch is tied to risk. You can plan everything from beginning till the end, but no matter how well you plan it out, things might just not work out.

It's just part of the job - in order to succeed, you need to take launch risky campaigns on a regular basis, and hopefully, one in every 5 is going to bring you massive results."

#### **Sample Answer 2:**

"I'm not a risk-taker, I'm more of a risk-manager. As someone who's been in finance for years, I can say with a lot of confidence that there's risk in everything.

The most important things are to one, minimize your risks, and two, minimize potential damages if everything goes very, very wrong.

While working at Investment Bank X, we had a very interesting policy for investing in new fintech projects. We used to avoid moon-shots, high-tech projects, as well as anything that had an experimental business model.

Our strategy was to invest in proven tech. As in, proven product-market fit, business model, etc. In most cases, these were runner up companies. We wouldn't invest in that one innovative company that was all over the news - we'd instead invest in their latest competitor. More often than not, this ended up being more profitable, and significantly less risky."

## 13) How do you deal with pressure or stressful situations?

If you're applying for a high-stress job, you're guaranteed to get asked this question.

The aim of this question is to see if you're the type of person who'd survive working the job, or fall through the cracks when the first signs of trouble show up.

Obviously, you wouldn't answer with the following...

*"Well, I end up having a panic attack, crying, and running away from work."*

Instead, answer as follows...

- Say that yes, you do tend to perform well during stressful situations
- Give 1-2 examples of a situation where you had to perform well under pressure

Now, let's go through some real-life examples:

#### **Possible answers:**

**Sample Answer 1:**[Register](#)[Sign In](#)

*"Though I can't particularly say I enjoy stressful situations, I AM very good at working under pressure.*

*During chaos and panic, I tend to take a step back, think, plan, and prioritize.*

*For example, there have been times I've had to juggle multiple university projects and assignments at the same time. I would break up large assignments into small, individual tasks, and prioritize based on:*

*How fast I could complete each task*

*Figuring out which task would take the longest*

*Which project had the earliest deadline*

*This way, my work became a lot more manageable. The most times I had to experience such situations, the better I performed overall."*

**Sample Answer 2:**

*"I actually prefer working under pressure. I look at it as a challenge - a situation where I really have to up my game to succeed.*

*As a cook, working under pressure is pretty much part of the job. I've been in several situations where the restaurant was understaffed for the occasion. Heck, it's pretty much a constant thing during peak season.*

*When there's a ton of orders coming in and we can barely keep up, I tend to get significantly more productive than usual."*

## 14) Do you prefer hard work, or smart work?

By definition, hard work is when you, well, work hard. It's when you're willing to put a lot of work to get the job done.

Smart work on the other hand, means doing the work efficiently. If you manage to get the job done in 2 hours instead of 5, with the same end-result, you're doing smart work.

Keep in mind, though, that by asking this question, the interviewer is looking to understand what your work ethic is like. Meaning, they're looking for a healthy combination of both, not just one.

That is, they want you to be the candidate who not only thinks smartly but works as well.

So, your answer here shouldn't be one-sided...

*"Oh, I loooove smart work. That's when you come up with what to do, and make others people do it, right?"*

Instead, explain how you excel at both:

**sample answer:**[Register](#)[Sign In](#)

*"I don't particularly have a preference - I believe that both hard and smart work is important to get the best results."*

*Smart work, on one hand, lets you figure out the best and most efficient way to get things done.*

*Hard work, on the other hand, means that you'll do the job right. Even if there's no way to do it smart or efficiently, you'll be willing to put in long hours of work to get it done.*

*I'm the type that does both.*

*For an example of smart for, during my time at [Made Up Corporation], I was in charge of the sales department. As a process improvement initiative, I migrated from an outdated, in-house CRM, to Pipedrive. This improved the department's productivity by around 20%.*

*On the other hand, the whole migration process took around 3 months of hard work. As the software we were using was outdated, trying to learn how to map and migrate our data was a lot more complicated than we'd expected."*

## 15) How quickly do you adapt to new technology?

Today, whether you're applying for a software engineering job, or as a cashier in a supermarket, you're going to need to use technology at least on some level.

It's very common for a company to adopt new tech - new point of service system check-out kiosks, customer management software, and whatever else.

So, you should be able to pick up **new tech ASAP**. Any new change shouldn't completely disrupt your work.

So, when answering this question, you should talk about how tech-savvy you are.

### Possible answers:

#### **Sample Answer:**

*"I'm pretty tech-savvy. I've worked with a lot of different Point of Service systems so far, and have zero difficulties learning how to use new ones."*

*As a given, I own a PC, have used Office 365, and all the usual stuff."*

#### **Sample Answer 2:**

*"I've always been interested in tech. In fact, I'm the type of person to actively seek out new software to help solve business problems at work."*



orked with 3 different Customer Management Systems such as PipeDrive, SalesForce, and Zoho CRM."

[Register](#)[Sign In](#)

## 16) Do you have any interests outside of work?

If the interviewer asks you this question, take it as a good sign!

It means that they liked your professional background, and now they're just trying to get to know you and see if you're a good fit for the company culture.

It's pretty hard to go wrong here, unless you're going to answer something like:

*"I have literally no hobbies."*

or...

*"All I do is play video games all day."*

Just talk about your [hobbies and interests](#), and you're all set!

Bonus points if you can mention something that's also relevant to your job (creative writing if you're applying for a copywriting job, for example).

### Possible answers:

#### Sample Answer 1

*"I'm a big fan of creative writing. I have my own personal short-story blog, and I contribute actively to several online writing communities (such as Writing Prompts on Reddit).*

*Oh, and I'm also a huge fan of the New York Giants."*

#### Sample Answer 2

*"Well, I'm very interested in all sorts of sports. I like to keep active, as it really helps keep me productive. Over the past 2 years, I've done a bit of everything from fencing, archery, hiking, and several other things."*

## 17) What do you think our company/organization could do better?

Well, this one's interesting!

While not too common for most organizations, it's a favorite amongst tech companies.



How come? Well, answering this question shows a couple of things:

[Register](#)

[Sign In](#)

- That you're really passionate about the organization, and have done your research
- Are not afraid of giving feedback

Obviously, you should be very political about your feedback. You can't just say the

### Incorrect example:

*"Well, a lot of things really. I'm not enjoying this interview right here, for example.*

*And your product kinda sucks, no offense. But hey, there's always room for improvement, am I right?"*

Instead, you want to show off the research you've done. Talk about anything that might seem off about their product or business:

### Correct example:

*"I actually went through your resume builder before coming to the interview, and found several things that seemed kinda counter-intuitive.*

*Not to say that it's too hard to understand, or something, but it took me a while to figure out some stuff.*

*If you want, I can open up my laptop and show you what I mean."*

## How to Answer 3 Most Common Behavioral Questions in an Interview [w/ Possible Answers]

Finally, **behavioral job interview questions** are questions that focus on how you've performed in the past, or how you'd perform in a specific situation.

They're used to understand your thinking patterns, and whether you can think on your feet.

Remember the [STAR interview method](#) we discussed all the way up? Well, it can be used to answer just about any behavioral interview questions.

To brush up your memory, the main idea behind the STAR method is that each answer should involve the following:



Question

Register

Sign In

- **T: Task** - Describe what your challenge or responsibility was.
- **A: Action** - List and dwell on all the actions you took towards addressing the challenge or responsibility.
- **R: Result** - Explain what the outcomes were and how they fit with the overall goal of the project or company.

Here are 3 of the most common behavioral interview questions (with sample answers):

## 1) Give an example of how you have handled a challenge in the workplace before.

What the interviewers want to know in this case is how well you handle conflict and difficulties.

So, the answer here should be pretty straightforward. You should describe a challenge you faced at work, and explain how you solved it.

### Possible answers:

#### Sample Answer 1

*"During my last job as a Google Ads expert, one of the company clients had accidentally butchered their own ad account. I noticed this over the weekend and saw that they were basically wasting money on nothing.*

*I took the initiative and immediately got in touch with the client to let them know about the issue. Then, we set up an impromptu meeting on the same day and fixed the account before any real damage could be done."*

#### Sample Answer 2

*"During my last job, I was managing the creative and web dev team in charge of creating an online store for a client. Two days before deploying, we found a major bug that messed up the whole front-end user experience.*

*Now, we could have pushed the deadline a bit, but that would have messed up the relationship with the client. The project was already postponed once because of unforeseen circumstances, so this one was a do or die.*

*I assembled a task force consisting of web developers from my team, as well as some software engineers from another department. We focused 100% of our time on fixing the issue, and actually managed to launch on time at the end."*

## 2) Give an example of when you performed well under pressure.

For any high-stress work environment, you're guaranteed to get asked this question.



Register

Sign In

## Possible answers:

### Sample Answer 1

*"I actually perform a lot better when under a lot of pressure. The sense of urgency and importance really motivates me to up my game and make sure everything works out right."*

*When I worked as a Financial Analyst at [MadeUpFinanceCompanyInc], most of the work was very high pressure. I had to go above and beyond the line of duty to make sure we met the tight deadlines set by our clients. This often meant working 12 hour work days, and sometimes, working over the weekend.*

### Sample Answer 2

*"As a seasonal worker, my entire career is high-pressure, haha... My last position was as a line cook during the summer at the SomeRestaurant in JacksonHole, Wyoming.*

*Around 3 out of 4 of the months were super high stress - there was a LOT of work, and the restaurant was pretty much always full. Heck, I've even had to skip breaks just to make sure we wouldn't be understaffed."*

## 3) Give an example of when you showed leaders qualities.

Applying for a leadership position? You'll definitely get asked this behavioral interview question.

Keep in mind, though, that this question doesn't necessarily mean that you should have held a managerial position.

What the interviewer is asking for is a situation when you took the initiative and led a project or an initiative.

### Possible answer:

*"As an entry-level marketer, there was not much expected of me at SoftwareCompanyInc.. My main to-dos involved doing research, and completing whatever tasks were assigned to me."*

*During a content marketing brainstorming session, I came up with an awesome idea to market the company. The gist of it was, we'd interview company clients who were very successful at using the software, create case studies on what exactly they're doing, and include it in our email marketing strategy.*



Chief Marketing Officer loved the idea of my project, which I did with flying colors.

[Register](#)[Sign In](#)

## **[BONUS] Job Interview Infographic**

Now, it's time for a quick recap! Check our our new infographic on some of the most common job interview questions (and how to answer them):





Register

Sign In

11



42

# Common interview **questions** and best **answers**



## 1. Why would you like to work here?

Mention what you like about the company and the position and how you can be a good addition to the work environment. This will show the interviewer that you have knowledge of the company and the job.



## 2. Why should we choose you?

Be confident when responding to this question. Have a short focused sales pitch where you say what you can offer the employer and how you can contribute to the success of the company.



## 3. Tell me about yourself

Share the right amount of personal information without getting into detail, especially about religious and political beliefs as they might cause contradictions. Talk about your education and motivation factors.



## 4. Where do you see yourself in 5 years?



Register

Sign In

This is a tricky question. They are trying to figure out if you are going to run after the first better opportunity. It's best if your answer is focused on the company and the job you are applying for.



## 5. What are your strengths?

Discuss the skills and competencies that qualify you for the specific position and make you stand out from the other candidates. Prepare examples that will match your strengths to the job requirements prior to the interview.



## 6. What are your weaknesses?

Try to answer by turning weaknesses into positive experiences that made you a better professional. Frame your response around your skills and qualifications.



## 7. How do you handle stress?

A good way to answer this question is to tell a story of how you handled a stressful situation at your previous job. An example would be if you were given a difficult task with a short deadline and you managed to complete it.



## 8. What motivates you?

With this question the interviewer is trying to find out if your personal motivations match the company culture and the Job specifications. The best strategy is to be honest and tell what would excite and motivate you if you were hired for the position.



## 9. What is your dream job?

Do your best to relate your answer to the job you are applying for. A way to respond to this question is to mention a certain personal or professional goal you'd like to reach through your "dream job." Avoid mentioning specific titles and saying "this job" as it might sound insincere.





Register

Sign In

## 10. What salary do you expect?

Remember, only offer a number that you will accept and be able to live with. Research salaries for similar positions. A safe answer would be "I would like to be compensated according to my qualifications and experience."



## Additional Interview Preparation Tips

### 11. Why did you leave your last job?

Another tricky question where you should stick to the facts and focus your answer more on the future. The good response would give the impression that you are motivated by the opportunities provided rather than trying to run away from an unpleasant job.



 novoresume.com

Follow basic interview etiquette, make eye contact with the interviewer, say 'please' and so on.

And finally, good luck on the interview! You got this!

Suggested readings:

- [How to Make a Resume & Land That Job](#)
- [How to Make a Cover Letter](#)
- [How to Ace a Video Interview](#)
- [Thank You Email After Interview](#)
- [Phone Interview Questions & Tips](#)

[Rate this article](#)

1340 Reviews



Author

**Andrei Kurtuy** – Resume, CV and Cover Letter Writing Expert

Interviewers want:

- To know about the journey that led you right to their doorstep.
- A VIP pass to your past performances, seeking evidence of how you shone in real-world situations.
- To ensure that your toolbox is well-stocked with the necessary technical and **soft skills**.

And hey, let's not forget: an interview isn't just a solo performance – it's a dialogue. To truly prepare for your job interview, arm yourself with some smart, insightful questions of your own. Whether you're chatting with a recruiter or your potential new employer, the questions you ask can shine a light on your enthusiasm and strategic thinking.

Lastly, remember that it's not only important to consider the questions recruiters might ask you or you might ask them, but also to be aware of **questions that should not be asked**. Knowing how to recognize and handle illegal interview questions ahead of time can make an awkward situation much easier to navigate.

To learn how to answer common interview questions and discover over 20 good questions to ask in an interview, start right here.

## Behavioral Interview Questions

These questions begin with phrases like "**Tell me about a time when...**" or "**Give me an example of...**". To tackle **behavioral interview questions**, paint a picture of how brilliantly you've handled previous challenges and sprinted toward success.

Use the **STAR Method** to give a better structure to your stories and share your experiences like a pro storyteller. Now, let's dive into the examples.

### 1. How Do You Handle Stress and Pressure?

If feeling the heat of an intense workday is part of the gig you're after, interviewers will most likely toss this question your way. They want to find out how your cool-headed strategies under fire can keep the team's spirits from going up in smoke.

**Example of how to answer this interview question:**

- *"I thrive under pressure, which I believe is as much about mindset as it is about action. For instance, during a critical product launch, I faced unexpected glitches. Instead of panicking, I prioritized my tasks, communicated transparently with my team about the challenges, and together we worked through the issue systematically. This approach not only resolved the problems efficiently but also strengthened our team's problem-solving skills for future projects."*

### 2. Describe a Time You Faced a Significant Challenge at Work.

Brace yourself for a question that's as common as coffee breaks in the workplace: describe a significant challenge you've faced. Interviewers ask this question to reveal how you turned obstacles into springboards for success.

**Example of how to answer this interview question:**

- *"Last year, our team was on the verge of missing a critical deadline for a new client, which could have derailed the entire project. I spearheaded an emergency plan that involved reassigning tasks based on each team member's strengths and streamlining our communication process. It was challenging, but by fostering a collaborative environment and keeping morale high, we delivered quality work on time. This experience taught me the importance of adaptability and clear communication in crisis management."*

### 3. Can You Provide an Example of When You Showed Initiative?

Let's talk about that spark of self-starter energy that can be the difference between a good employee into a great one. When interviewers ask you to share a time you showed initiative, they're on the lookout for a self-motivated person who doesn't wait for a green light to accelerate toward improvement and success.

**Example of how to answer this interview question:**

- *"There was a moment when I observed that many customer queries were about the same few issues, which was overwhelming our support staff. Without being asked, I took the initiative to create detailed FAQs and troubleshooting guides. I then collaborated with the product and customer service teams to ensure accuracy and accessibility. After we published these resources, customer complaints decreased by 25%, and our resolution time improved significantly."*

### 4. Talk About a Time When You Had to Work Closely with Someone Whose Personality Was Very Different From Yours.

Teamwork makes the dream work, but what if your teammate is more of a solo artist? Recruiters ask this question to learn how you interact with diverse personalities in the workplace. This is important because the ability to click with different people is what transforms a group of individuals into a power squad.

**Example of how to answer this interview question:**

- *"In my previous job, I partnered with someone who was incredibly detail-oriented, while I'm more of a big-picture thinker. Initially, our approaches clashed, but then I realized it was a gift-in-disguise. I proposed we divide tasks to play to our strengths. I would outline the project's vision, and my coworker would fine-tune the details. This synergy not only improved the project outcome but also our mutual respect and work relationship."*

### 5. Tell Me About a Time You Failed. How Did You Deal With the Situation?

Oh, the dreaded F word – Failure. But in an interview, think of it as F for fortune because this question is a golden opportunity to showcase your resilience and learning agility. Interviewers want to see that you've got the determination to get back up after a stumble.

**Example of how to answer this interview question:**

- *"In my role as an **event coordinator**, I once overlooked a critical vendor booking for a large conference. Although I tried to secure a last-minute replacement, unfortunately, the booking had to be postponed due to my error. However, to ensure this never happened again, I conducted a thorough investigation to understand the oversight and implemented a double-confirmation system for all future bookings. From this humbling experience, I learned the importance of meticulous cross-checking in event management."*

## 6. Discuss a Time When You Went Above and Beyond the Call of Duty.

Here's your chance to tell a cool story. One where you are a superstar who doesn't just do the job to a minimum but rises above. Show recruiters you are the type of employee who throws an extra scoop of awesome into the work sundae just because you can.

**Example of how to answer this interview question:**

- *"As an **assistant manager**, I noticed the team's morale was low following some economic layoffs. I initiated a weekly 'Good News' session, where everyone shared positive achievements, both personal and professional. It wasn't in my job description, but this effort boosted team spirit and, in turn, productivity. It showed me how little acts of leadership can have a big impact on workplace culture."*

## 7. Provide an Example of a Goal You Reached and Tell me How You Achieved It.

Got goals? We bet you do. And when an interviewer asks you about them, they're on a hunt for evidence of your drive and ability to turn plans into reality. They want to know if you've got the vision to set sail towards ambitious horizons, plus the methodology and commitment to achieve results.

**Example of how to answer this interview question:**

- *"In my previous role as a **content manager**, I aimed to increase our website's traffic by 30% within six months. Diving into keyword research, revamping our SEO strategy, and pioneering a series of collaborative guest posts, I tracked our performance meticulously. This hands-on adaptive approach paid off with a 45% traffic surge, surpassing our original goal and proving the power of a targeted content strategy."*

## 8. Tell Me About a Time When You Had to Deal With Conflict on the Job.

This **conflict-resolution question** is a recruiter's way of taking a sneak peek into how you navigate workplace drama. When answering this question, show off your soft skills. Recruiters will

be looking to evaluate your emotional intelligence, problem-solving abilities, and talent for maintaining professional composure under fire.

#### Example of how to answer this interview question:

- *"In my last role as a **customer service manager**, two team members clashed over how to handle a recurring problem with a product. I mediated the situation by arranging a meeting to discuss the issue openly. By encouraging a solution-focused dialogue, we identified the root cause and agreed on a customer-centric solution that satisfied everyone. This incident underscored the importance of open communication and teamwork in resolving conflicts."*

## 9. How Do You Handle a Situation Where You Have to Meet Multiple Deadlines?

Deadlines are the bread and butter of workplace productivity. Recruiters ask this question to assess your time-management skills. It's their way of figuring out if you can tackle the tick-tock of the clock and juggle, prioritize, and triumph without breaking a sweat.

#### Example of how to answer this interview question:

- *"When faced with simultaneous project deadlines, I lean on my prioritization skills. For instance, at my last job, I organized tasks by urgency and impact, allocating time to each based on their deadline and importance. I managed to submit all projects on time without compromising on quality."*

## Situational Interview Questions

Think of these questions as a flight simulator for job candidates, where hiring managers can witness your piloting skills before you ever leave the ground. Situational job interview questions and answers like the following give your potential employer a preview of how you'd handle the twists and turns of real-life work situations. They want to test your **problem-solving**, critical thinking, **decision-making**, and adaptability skills.

## 10. How Would You Manage a Project With a Tight Deadline?

Can you juggle all the balls and keep them from falling? When interviewers ask you this question, they want to know about your project management skills. They're testing your organizational acumen, prioritization skills, and ability to remain calm when the clock is ticking ominously in the background.

#### Example of how to answer this interview question:

- *"When faced with a tight deadline, I immediately break down the project into manageable tasks and set mini-deadlines for each. For example, in my last role, I had to deliver a marketing campaign in two weeks. I delegated tasks according to my team's strengths, set up daily progress meetings, and we managed to complete the project two days early, with results that exceeded our targets."*

## 11. How Would You Handle a Difficult Client or Customer?

This is one of the recruiters' top interview questions when scouting for diplomacy skills. They most likely want a peacemaker who can turn tension into achievement. How you steer these difficult interactions says a ton about your patience, empathy, and customer service prowess.

**Example of how to answer this interview question:**

- *"I handle tough client situations by first actively listening to their concerns without interrupting. For instance, when addressing a former client's frustration over a delayed order, I apologized sincerely, explained the unforeseeable delay, offered a discount, and expedited shipping. The client was so pleased with the resolution, that they placed another order on the spot."*

## 12. What Would You Do If You Disagreed With the Way a Manager Wanted You to Handle Something?

Disagreeing with a boss is a delicate business - a dance between respect and assertiveness. Interviewers ask this question to assess your ability to navigate disagreement with tact, to be professional yet confident. They want to test your conflict resolution skills and see if you can disagree without discord, championing collaboration over confrontation.

**Example of how to answer this interview question:**

- *"If I disagreed with my manager's approach, I'd request a private meeting to discuss the problem openly. I'd preset my concerns clearly and respectfully, backed by concrete data or examples. For example, when I felt an alternative vendor could offer better service, I prepared a comparison report to discuss with my manager, which ultimately led us to a better partnership decision."*

## 13. If You Saw a Coworker Doing Something Dishonest, What Would You Do?

Honesty and integrity are the bedrock of a trustworthy employee. When interviewers ask this, they're probing your ethical standards, and assessing your judgment in handling sensitive situations.

It's not just about doing what's right but showing leadership even when you're not in charge. Recruiters want to see that you can balance personal and professional integrity without resorting to creating office drama.

**Example of how to answer this interview question:**

- *"If I observed a coworker engaging in dishonest behavior, I would follow the company's established protocols for such incidents. This typically means first trying to understand the situation more fully – perhaps the coworker is unaware that their actions are inappropriate. If a simple conversation does not resolve the issue, or if the dishonesty is significant, I would then report the matter to my manager or the appropriate department, like HR. I believe it's*

*important to handle such situations sensitively and confidentially to maintain a positive and ethical work environment.”*

## 14. If You Could Choose Any Company to Work For, Where Would You Go and Why?

If you're asked **questions about your ideal company**, it's a clever way for interviewers to peek into your career aspirations, values, and alignment with their company's culture and mission. It reveals much about what you prioritize in a job and if you're likely to stick around if they hire you.

Example of how to answer this interview question:

- *“I would choose Monster. The platform’s dedication to facilitating meaningful employment matches appeals to my passion for helping others achieve their professional goals. Working with Monster would allow me to be at the forefront of transforming job hunting into a more intuitive and satisfying experience. Their innovative approach to career development services aligns with my ambition to be part of a team that values impactful guidance and empowers job seekers through technology and resource accessibility.”*

## 15. How Would You Prioritize Your Work If You Had Multiple Tasks With the Same Deadline?

Balancing multiple tasks in a high stake environment is your jam, but how exactly do you decide what to do first and what to delegate or reevaluate? Explain your process of analyzing and organizing tasks based on urgency and importance. This is a chance to show recruiters that you thrive in a fast-paced environment.

Example of how to answer this interview question:

- *“In such situations, I prioritize work that is critical to customer satisfaction or operational continuity first. For example, in my last role, I found myself having to prepare a client’s order, finalize a quarterly report, and update the inventory all by the end of the day. I delegated the inventory task to a trusted colleague, completed the client order first due to its immediate revenue impact, and then compiled the report, as it had a direct influence on our next quarter’s strategy.”*

## 16. What Steps Would You Take If You Were Assigned to Work With a Difficult Team Member?

With this question, recruiters want to observe your approach to teamwork, conflict resolution, and interpersonal communication. They are looking for signs of your leadership potential and emotional intelligence in the face of adversity. Prove your ability to maintain a positive work environment and share your conflict-resolution tactics.

Example of how to answer this interview question:

- *"I would start by having a one-on-one conversation to understand their perspective and express my concerns calmly. For instance, at my last job, I worked with someone who seemed resistant to group ideas. By discussing the matter privately, we realized it was a misunderstanding. He felt his ideas were not being heard. We agreed to be more collaborative in future discussions and developed a new feedback system. As a result, our team dynamics improved significantly."*

## 17. How Do You Stay Motivated When Your Job Requires Repetitive Tasks?

Employers want to know how you keep your spirits high on the job because motivation is the fuel that keeps the productivity engine running smoothly. Can you sustain performance over the long haul without needing constant external stimulation? This question digs into your self-drive and whether you can find purpose and passion even in routine work.

**Example of how to answer this interview question:**

- *"I maintain motivation by setting personal goals and benchmarks for my performance, even in repetitive tasks. For example, when I worked as a **data entry clerk**, I would challenge myself to increase my typing speed and accuracy. I also tried to find ways to improve the process itself, which not only kept me engaged but also added value to my team."*

## 18. What Would You Do If You Were Asked To Perform a Task You Have Never Done Before?

This interview question is a test of your resourcefulness and adaptability. Employers ask it to uncover your problem-solving skills and your willingness to tackle challenges head-on, even when the path isn't clearly mapped out.

**Example of how to answer this interview question:**

- *"In my previous role, I took on the challenge of creating a social media marketing strategy from scratch, a responsibility I haven't faced before. I started by researching effective strategies in our industry and learning from existing successful campaigns. Then, I outlined a plan, sought feedback from knowledgeable colleagues, and iterated on the strategy based on their insight."*

## 19. How Would You Handle Receiving Constructive Criticism From Your Manager?

This is one of those typical interview questions recruiters ask to separate flexible and forward-thinking candidates from those who may resist change and growth. Employers value employees who can take criticism in stride, using it as fuel for professional development rather than a hit to their ego. Think about this as an opportunity to demonstrate your maturity and dedication to continual improvement.

**Example of how to answer this interview question:**

- *"In my last position, my supervisor noted that while detailed, my reports were frequently too lengthy for our fast-paced meetings. I appreciated this feedback as it aimed to improve efficiency. I responded by creating a template for concise reporting, which my manager and the team adopted company-wide."*

## Culture Fit Interview Questions

Finding the right cultural fit means meshing well with a company's environment and values while contributing to your and the company's growth. Now, don't mistake 'cultural fit' with 'identical thinking'. Companies aren't looking for employee clones but individuals who can ride the company's wavelength while bringing their unique spice. It's about harmony, not uniformity. Showcase your values, work ethic, and social savvy in a way that says, "I belong here."

### 20. What Would Be Your Ideal Work Environment?

Employers ask the **ideal environment interview question** to understand your personal preferences and see if the office vibe will be music to your ears or just noise. Are you the missing puzzle piece in their workplace picture? They are fishing for clues to predict whether you'll fit, thrive, and stick around.

Example of how to answer this interview question:

- *"For me, the ideal environment balances autonomy with accountability. I appreciate having clear objectives and the freedom to figure out the best way to meet them, which was something I loved about freelancing. Yet, I also value a setting where feedback is frequent and performance is measured, ensuring we're all aligning with the company's vision."*

### 21. What Do You Know About Our Company Culture, and Why Do You Think You Would Fit In?

Always do your homework and **research the company before your job interview**. Recruiters will ask you this question to assess whether you've taken the time to get familiar with the company's values, work style, and atmosphere. This is your chance to demonstrate your alignment with the company's principles.

Example of how to answer this interview question:

- *"From my research and everything I have seen during the interview process, it's clear that your company values proactive communication and a team-oriented approach. I have always been proactive in my communications, often bridging gaps between different departments. I prefer environments where transparency and cooperation are the norm, and I am eager to contribute to and grow with such a team."*

### 22. Can You Tell Me About a Time When You Had to Adapt to a Significant Change at Work?

This common interview question is designed to assess your flexibility, your resilience, and whether you handle the winds of transformation with grace and efficiency. It's an insight into your problem-solving skills and your potential to grow with the company.

**Example of how to answer this interview question:**

- *"In my previous role, after our company's acquisition, we immediately shifted our project management tools and workflow. I took the initiative to organize group training sessions. Not only did I adapt to the new software quickly, but I also helped my team transition by creating how-to guides tailored to our projects."*

## 23. What Motivates You to Come to Work Every Day?

Are you in it just for the paycheck, or do you have a deeper connection with your work?

Recruiters use this question to assess whether you'll have the drive to push through on tough days and if your incentives align with the company's values and goals. Reflect on the aspects of your work that spark your enthusiasm.

**Example of how to answer this interview question:**

- *"Working towards a goal is what motivates me. The rush of meeting a target or exceeding expectations is thrilling. I thrive in a competitive atmosphere where my achievements can not only advance my career but also contribute to the company's success."*

## 24. How Do You Handle Working With a Team That Has Diverse Perspectives?

Diversity in the workplace is like assembling a vibrant, multifaceted mosaic— each unique piece contributes to a more compelling and complete picture in team projects. The question then becomes, how do you harmonize these diverse elements to craft a masterpiece that resonates with everyone, rather than creating a disjointed collage? Employers ask this question to see if you are the kind of team player who actively values and leverages differences to enhance creativity and innovation.

**Example of how to answer this interview question:**

- *"Working with a diverse team excites me because it's an opportunity to learn and grow. I make it a point to listen actively and ask questions that help me understand where others are coming from. This approach has helped me adapt my work style to suit different team needs. I'm eager to join your company because I know that you consider diversity a strength that drives success."*

## 25. What Is Your Approach to Maintaining Work-Life Balance?

A well-balanced employee is often seen as more productive, happier, and better at problem-solving. By showing your prospective employer that you have a solid game plan for maintaining

your work-life balance, you demonstrate self-awareness and that you are in it for the long haul – aiming for a sustainable and successful career path.

#### Example of how to answer this interview question:

- *"I believe in being proactive about my work-life balance. For instance, I schedule my tasks to prioritize efficiency during work hours and set clear boundaries for after-work availability. When I'm off work, I unplug completely and engage in activities like hiking or yoga, which help me return to work refreshed."*

### 26. How Do You Contribute to a Positive Team Environment?

Creating a positive team environment is like being part of a band – you have to tune in to others, know your parts, sometimes lead and other times follow, to make harmonious music. Interviewers ask this question to assess how well you play with others, your interpersonal skills, and if you are a team player who will contribute to the group's success and morale.

#### Example of how to answer this interview question:

- *"In my current role, I contribute to a positive team environment by maintaining open communication and a spirit of cooperation. I kick off projects with brainstorming sessions that welcome all ideas, ensuring everyone feels heard and valued. I also make it a point to recognize my colleagues' efforts, which I believe fosters a culture of appreciation and motivation."*

### 27. What Do You Expect From a Supervisor to Help You Perform at Your Best?

Employers often pose this question in interviews to uncover what management style you respond to best. It reveals your professional needs and expectations, as well as how you view the role of a leader in your work journey.

#### Example of how to answer this interview question:

- *"I expect a supervisor to be clear about expectations and available for guidance without micromanaging. I thrive when given autonomy to complete my tasks, knowing I can rely on my supervisor for support when challenges arise. For example, at my last job, my supervisor conducted regular but brief check-ins to keep us aligned, which I found incredibly effective for staying on track."*

### 28. Describe How You Have Contributed to a Team's Success in a Past Role.

Employers ask this to gauge how you define success, your role in achieving it, and whether you elevate the team's goals above your ambitions. Show them your collaborative spirit and your capacity to be a team player by pointing out specific instances where your contributions made a tangible difference.

### Example of how to answer this interview question:

- “As a **marketing analyst**, I contributed to my team’s success by identifying a key market trend that led to a 20% increase in lead generation. I conducted thorough research and shared my findings with the team, which then shaped our strategy moving forward. I also took the initiative to lead a series of workshops to keep my team updated on the latest market research techniques.”

## Leadership Interview Questions

This is your chance to shine like the beacon of guidance you are and demonstrate that you’re as wise as a council of Yodas. Leadership isn’t just about steering the ship – it’s about inspiring the crew, reading the stars, and sometimes swabbing the deck right alongside your team. When asked **leadership interview questions**, talk about your vision, strategy, decision-making, and how you react in tough situations.

### 29. Describe Your Leadership Style.

With this interview question, recruiters want to understand how your unique brand of leadership aligns with their team’s needs and the company’s values. It’s your chance to showcase how you inspire performance, handle challenges, and drive results – all while keeping the team’s spirit flying high. Use real-life examples to demonstrate how your leadership made a tangible impact where it matters most.

#### Example of how to answer this interview question:

- “I lead with empathy and support. I focus on understanding the individual strengths and motivations of my team members, which allows me to tailor my guidance to help them excel. By ensuring that team members have the resources and emotional support they need, I help to create a work environment that is nurturing yet result-driven.”

### 30. Give an Example of How You’ve Motivated Others.

Interviewers often ask this question to tap into your personal influence, emotional intelligence, and leadership potential. They want to see how you fuel the fire of enthusiasm in a work setting. It’s not only about what you’ve achieved but also how you’ve lifted others along your journey to success.

#### Example of how to answer this interview question:

- “In a previous role, I set a series of incremental goals that were attainable yet ambitious and motivated my team by recognizing individual achievements in our weekly meetings. With each success, the team’s enthusiasm and effort increased.”

### 31. How Do You Handle Delegating Tasks to Team Members?

When recruiters ask this question, they're looking for insights into how you ensure tasks are understood, supervised, and completed, without micromanaging or overloading your team members. Articulate your delegation process during the interview to underscore your strategic approach to leadership and team management.

**Example of how to answer this interview question:**

- *"I believe delegation is an opportunity to empower others. By involving the team in decision-making about who takes on which tasks everyone felt invested in the project. In the past, I've implemented this when planning a product launch and I also made sure to provide the resources and authority needed to accomplish all tasks, which promoted accountability and leadership within the team."*

## 32. Tell Me About a Time When You Had to Lead a Project With Little to No Direction.

This question reveals how you interpret objectives, set goals, and mobilize resources when the path isn't clearly marked. It also tests your resilience and problem-solving skills in situations where guidance is minimal. Your response should illustrate your capability to thrive in environments that require both autonomy and strategic thinking.

**Example of how to answer this interview question:**

- *"In my previous role as an event planner, I was tasked with developing a launch event for a new product. Without specific guidelines, I began by setting clear objectives with my team, establishing milestones, and brainstorming promotion strategies. This structured approach resulted in a successful event that exceeded our attendance goals by 25%."*

## 33. How Do You Evaluate the Success of Your Team?

Understanding how a candidate measures team success is vital for employers. It provides insight into the candidate's leadership style and accountability standards. This question also sheds light on the individual's commitment to continuous improvement and whether they value results or processes more.

**Example of how to answer this interview question:**

- *"I measure success not just by meeting deadlines and budgets but also by monitoring our team's growth over time. This includes analyzing our ability to reduce errors, increase efficiency, and enhance client satisfaction. Regular retrospectives help us learn from each outcome and refine our approach continually."*

## 34. Describe a Situation Where You Had to Resolve a Conflict Within Your Team.

Navigating team conflicts is a testament to strong leadership and problem-solving skills. When interviewers ask this question, they want to see how you handle pressure, mediate disputes, and

foster a collaborative environment. It's not about the conflict itself, but your approach to transparency, empathy, and finding solutions that maintain team cohesiveness and productivity.

#### Example of how to answer this interview question:

- *"When I noticed rising tensions due to uneven workloads, I initiated a team meeting to redistribute tasks. I introduced a shared tracking system for better visibility of everyone's responsibilities. This proactive approach not only alleviated stress but also increased overall team efficiency and morale."*

### 35. How Do You Foster Teamwork?

Teamwork is the glue that holds tasks, people, and goals together in any thriving workplace. Recruiters ask this question hoping to get insights into your collaborative skills and your role in building a positive team dynamic. They want to understand how you encourage participation, support a shared vision, and create an inclusive environment that leverages diverse strengths for collective success.

#### Example of how to answer this interview question:

- *"To foster teamwork, I maintain open lines of communication and ensure that all team members feel heard. For instance, I introduced weekly meetings where team members could bring up any ideas or issues. This practice not only prevented potential conflicts but also affirmed that we valued and considered every voice critical to our shared mission."*

### 36. What Strategies Do You Use to Make Decisions Under Pressure?

Decisions are the steering wheel that directs the course of business action and, under pressure, that wheel can either hold steady or veer off course. Do you possess the composure, analytical skills, and foresight needed to handle tough calls when the stakes are high?

#### Example of how to answer this interview question:

- *"When under pressure, I prioritize the tasks at hand based on urgency and impact. For instance, when facing tight deadlines on two major projects, I assessed which project had the potential to affect more stakeholders and prioritized my team's focus accordingly, ensuring we met our goals without compromising quality."*

## Career Goals and Aspirations Interview Questions

This is the moment in your interview where the spotlight shines on your future vision, revealing where you aim to go and how you plan to get there. These aren't random interview questions, but a vital tool for recruiters to unveil if there's any synergy between your career trajectory and the company's direction. Share your professional desires, ambitions, and the steps you're eager to take toward achieving greatness.

### 37. Where Do You See Yourself in Five Years?

This question is a recruiter's way of checking if you have a forward-thinking mindset and ambition to grow alongside them, or if you're just a shooting star, passing briefly through their galaxy. Check out our variety of tips on how to [answer "where do you see yourself in five years?"](#) so you skillfully navigate this common interview question with confidence and clarity.

**Example of how to answer this interview question:**

- *"I envision myself in a senior role, contributing to impactful policy change within the sector. Working with an organization that values social responsibility, like this one, I aim to leverage my skills in advocacy and program development to make a substantial difference in the community."*

## 38. How Do Your Career Goals Align With This Position?

Recruiters want to know that their position isn't just a temporary parking spot on your career highway but a meaningful milestone for you. Prove that you've done your homework on where the company is going and that your career compass points in the same direction as their mission.

**Example of how to answer this interview question:**

- *"My career goal is to become an expert in **graphic design**, and this position is a perfect match. Your company's innovative approach to branding is something I have always admired. I am excited about the prospect of bringing my unique style and fresh ideas to your team, helping to push creative boundaries and drive brand engagement."*

## 39. What Is Your Dream Job, and Why?

Recruiters ask this question to see if your values and ambitions align with the job at hand. Practice interview questions like this one to tie your ultimate career wishes to the realities of the position you're interviewing for. Get their attention by showing commitment and clarity.

**Example of how to answer this interview question if interviewing for your dream job:**

- *"Actually, this position is my dream job. I thrive on challenges like those that characterize your company. I have always wanted a role that blends my passion for cutting-edge technology with my desire to make a real-world impact, which is what this role is all about."*

**Example of how to answer this interview question if your dream job is different:**

- *"My dream job is to lead a software development team because it combines my love for coding with my passion for team building and mentorship. While this role as a **software developer** is a step on that path, I believe your company's culture of leadership development and technical excellence is the perfect environment in which to improve those skills."*

## 40. Describe the Steps You Have Taken Towards Professional Development.

Companies love growth-oriented go-getters, and they might ask you this question to know if you are as invested in your own progress as they are in their success. Your answer should demonstrate initiative, commitment, and a genuine interest in self-improvement. It should outline specific actions you've taken to learn new skills, advance your knowledge, and stay up-to-date in your field.

**Example of how to answer this interview question:**

- *"In pursuit of becoming an **SEO content manager**, I have completed certifications in SEO and analytics, and I have attended writing workshops to sharpen my craft. I have also led a content campaign as a volunteer, increasing web traffic for a nonprofit by 40%."*

## 41. What Professional Achievements Are You Most Proud of?

This question unpacks your definition of success and shows what you value in your work. It can reveal your strengths, aspirations, and fit for the company culture. A great answer to this question should go beyond listing achievements by connecting them with real value delivered to previous employers or your own development.

**Example of how to answer this interview question:**

- *"The professional achievement I hold in the highest regard is developing a new product feature that became the top revenue generator for my current company. This experience enhanced my product development and market analysis skills, which I plan to expand upon in the more strategic role I seek now."*

## Problem-Solving Interview Questions

These common interview questions are like the intellectual equivalent of an obstacle course, asking you to demonstrate agility, strength, and strategy all at once. This is your moment to show off your critical thinking, creativity, and cool-under-pressure demeanor. It's your chance to show recruiters that you can easily transform a "Houston, we have a problem" moment into a "Ta-da! Watch me fix it" showstopper.

## 42. Can You Describe a Time When You Had to Solve a Difficult Problem at Work?

When recruiters ask this question, they want to know how you approach a problem, weigh your options, and follow through. Your answer should paint a picture of the problem and your resolution journey, including the skills you applied, the steps you took, and the outcome. Emphasize your analytical abilities, creativity, and tenacity without sacrificing the wisdom of seeking help when needed.

**Example of how to answer this interview question:**

- *"Once, a product launch I was managing was at risk due to a critical supply chain disruption. To solve this, I worked closely with the operation team to identify alternative suppliers. We*

*negotiated expedited shipping to meet our deadline. As a result, the launch was not only on time but came under budget, strengthening our team's adaptability and vendor relationships."*

## 43. How Do You Approach a Situation Where You Don't Have All the Answers?

This is your chance to prove that you're not one to shy away from the occasional "I don't know". Instead, you are the one who will find out, pulling knowledge and expertise from wherever it lies. With this interview question, recruiters attempt to assess your humility, learning attitude, ability to leverage resources, and how you collaborate with others.

**Example of how to answer this interview question:**

- *"If I'm confronted with a question from a student that I can't answer, I see it as a learning opportunity for the class and myself. I take a 'let's research together' approach where we ultimately collectively learn much more than if I had provided an immediate answer."*

## 44. Tell Me About a Creative Solution You Have Designed to Address a Work Issue.

Interviewers often ask this question to see the originality of your problem-solving skills, if you can think outside the proverbial box, and how your unique solutions can bring value to the team and the company. When answering, highlight your ability to identify challenges, your creativity in solving problems, and the impact of your solution.

**Example of how to answer this interview question:**

- *"I noticed our team's response time to customer inquiries was slow due to an outdated ticketing system. I proposed and led the adoption of an AI chatbot that could handle common questions and free up our reps for complex issues. This solution cut our average response time in half and increased customer satisfaction scores by 15%."*

## 45. How Do You Prioritize Issues When Multiple Problems Arise Simultaneously?

This question is recruiters' attempt to understand your problem-solving hierarchy. The best answer to this interview question should highlight your ability to assess situations quickly, manage time effectively, and communicate priorities clearly.

**Example of how to answer this interview question:**

- *"In my previous role as an **IT specialist**, I often had to deal with simultaneous system failures. I prioritize based on impact and urgency. For instance, if two issues occur at once, like a server outage and a printer jam, I tackle the server first because it affects all users company-wide. My goal is always to minimize downtime and maintain productivity."*

## 46. How Do You Assess the Risks When Making a Decision?

Evaluating risk is an essential part of decision-making, especially in roles that affect the trajectory of a business. Interviewers ask this question to determine if you have a strategy for anticipating, understanding, and managing potential setbacks in a project or business decision. Explain how thorough you are with your homework, paying attention to all tiny warning signs.

**Example of how to answer this interview question:**

- *"When I'm at the decision-making helm, I start by charting out the 'what-ifs' and considering all possible outcomes and their implications. I evaluate past projects for unforeseen challenges and seek advice from subject matter experts to understand potential future conditions. For example, in a project with significant consequences, I developed alternative plans to address various possible issues that could arise, guaranteeing continuity of operations despite unexpected difficulties."*

## 47. Tell Me About a Time You Made a Mistake in Your Problem-Solving Process and How You Corrected It.

Interviewers use this question to get a sense of your humility, resilience, and how you handle setbacks and challenges. They look for candidates who can demonstrate growth and the ability to learn from their errors. The best answer to this question conveys self-awareness, responsibility, problem-solving skills, and an understanding that mistakes are a valuable part of learning and growth.

**Example of how to answer this interview question:**

- *"During a software rollout, I focused too much on meeting the launch date rather than ensuring full functionality. When early users encountered bugs, I owned up to the oversight and led my team in a user-focused testing phase, which not only fixed the bugs but also gave us deeper insights into user needs, improving the final product."*

## 48. How Do You Ensure That a Solution is Both Effective and Efficient?

In the fast-paced rhythm of today's workplaces, it's not enough for a solution to just work. It has to work smart. Interviewers ask this question to see if you have a good eye for solutions that strike an optimal balance between quality and resourcefulness. The ideal answer should illustrate your ability to think critically, streamline processes, and measure outcomes.

**Example of how to answer this interview question:**

- *"I believe in regular audits of the solutions we implement. For instance, after overhauling an inventory system, I scheduled monthly reviews to track various performance metrics against our goals. This approach has consistently helped us fine-tune the system to adapt to changing demands while ensuring optimal resource usage."*

# Communication Skills Interview Questions

Strong communication is what connects the many moving parts of an organization. Whether you're the voice that calms the storm, the email that lights up a smile, or the presentation that sparks inspiration, your ability to convey ideas clearly and effectively is crucial. These questions give you a chance to prove that you're not just a talker but a communicator whose every word leaves a lasting impression.

## 49. How Would You Describe Your Communication Style?

With this question, interviewers are trying to find out if your style of communication will harmonize with the company's culture. Explain how you interact with colleagues and clients, manage conflicts, and ensure your message is understood and appreciated. Your answer should demonstrate clarity, adaptability, and ability to balance speaking and listening.

**Example of how to answer this interview question:**

- *"I prioritize empathy in my communication, striving to understand others' viewpoints before responding. This approach helped me successfully mediate a conflict between two departments, by ensuring each side felt heard and valued before moving towards a resolution."*

## 50. Describe a Situation Where You Had to Explain a Complex Concept to a Coworker or Client.

Unraveling complex concepts for others is like translating a foreign language into a local dialect – it takes skills, patience, and understanding. When interviewers ask you to recount such a situation, they're scanning for your ability to make the intricate simple and your talent for guiding others. Show recruiters that you can dissect big ideas into bite-sized, digestible pieces.

**Example of how to answer this interview question:**

- *"When our company introduced new project management software, I was responsible for training my non-tech-savvy colleagues. I broke down the software functionality into a series of step-by-step tutorials and used relatable analogies to familiar processes. By comparing the software workflow to a well-organized filing cabinet, colleagues quickly grasped the concept, which led to a smooth transition."*

## 51. How Do You Handle Giving Negative Feedback to a Team Member?

Let's face it: no one throws a party when it's time to give negative feedback. But as a leader or teammate, it's part of the gig. Remember, it's not just about being honest or direct but also about being constructive and kind, ensuring that feedback leads to growth, not resentment. Highlight your emotional intelligence, professionalism, and commitment to fostering a positive team environment, even when things get tough.

**Example of how to answer this interview question:**

- “When a team member’s performance was slipping, I scheduled a private, informal chat. I used the ‘sandwich’ feedback technique, which starts and ends with positive notes. I started by acknowledging their past contributions before describing the current issue. Together, we developed an improvement plan, and I made it clear I was there for support. This turned a potentially tough conversation into a collaborative effort for better results.”

## 52. Tell Me About a Time When You Successfully Persuaded Someone to See Things Your Way.

How do you harness the power of persuasion in a professional setting? Being persuasive is about dialogue, understanding, and respect. This question tests your ability to communicate compellingly, negotiate, and win others over with logic and empathy, not just passion.

Example of how to answer this interview question:

- “In my previous **sales role**, I persuaded our team to adopt a new customer relationship management system. I demonstrated how it could save us time and increase leads by showcasing a successful case study and running a cost-benefit analysis. My methodical approach helped the team see the long-term benefits, leading to a unanimous agreement to make the switch.”

## 53. What Is Your Approach to Handling Customer Complaints?

Recruiters use this question to assess your ability to listen, empathize, and act effectively, ensuring customer satisfaction and loyalty while upholding the company’s reputation. Prove to them that you can maintain poise under pressure and turn a potentially negative experience into a positive one.

Example of how to answer this interview question:

- “When faced with a customer complaint, I listen actively to fully understand the issue. For instance, when a customer reported a problem with a product, I listened without interruption, apologized for the inconvenience, and offered a replacement along with a discount on their next purchase. This approach resolved the issue and helped retain the customer.”

## 54. How Do You Deal With Miscommunication in the Workplace?

Miscommunication can sneak into the workplace like an uninvited guest, and it can disrupt the flow of even the smoothest-running operations. Interview questions like this one give you the chance to showcase your ability to clarify, collaborate, and correct with a level head and a focused approach.

Example of how to answer this interview question:

- “When I detect a miscommunication, my first step is to address it directly with the involved parties. For example, when a project brief was misunderstood by my team, leading to a mismatch in project outcomes, I organized a meeting to discuss the confusion, realigned our

*understanding, and established clearer channels for ongoing communication to prevent future issues."*

## 55. What Strategies Do You Use to Communicate Effectively in a Team?

Effective communication is an essential skill that includes knowing how to articulate ideas, listen to feedback, and foster an environment where information travels without getting lost in translation. Explain to recruiters how you make sure communication in your team is a two-way street and everyone feels heard and empowered to contribute to the team's success.

Example of how to answer this interview question:

- *"I focus on clarity and simplicity in my communications. This means breaking down complex tasks into actionable items. Once, I converted a complicated project workflow into a simple checklist, which my team used to complete the project ahead of schedule with fewer errors."*

## 56. Tell Me About a Time When Your Communication Skills Improved a Situation.

Interviewers ask this question to assess your interpersonal abilities and witness the transformative power of your words firsthand. They're interested in how you use communication as a tool to navigate and enhance workplace dynamics, solve problems, and drive positive change.

Example of how to answer this interview question:

- *"There was a time when miscommunication between two departments led to a standoff that affected project delivery. I stepped in as a neutral party, arranged a meeting to facilitate dialogue, and provided clear, structured mediation. By the end of our session, not only had we found a resolution, but the teams also agreed to a new protocol for future collaboration, enhancing cross-departmental harmony and productivity."*

## Creativity and Innovation Interview Questions

During a job interview, expect employers to poke at the boundaries of your creative mind, looking for evidence that you can bring fresh perspectives to solve old problems. This is your chance to showcase that you're the kind of innovator who doesn't just follow trends – you set them. Whether it's about reimagining a process, dreaming up a product, or leading a revolutionary campaign, you'll need to show that your creativity is a powerful engine for real-world innovation and change.

## 57. Describe a Time When You Had to Think Outside the Box to Solve a Problem.

A great answer to this question goes beyond just a unique solution. You must demonstrate your innovative thinking, flexibility, and willingness to experiment with new approaches to finding a solution.

**Example of how to answer this interview question:**

- “Confronted with a limited budget for a marketing campaign, I turned to social media to crowdsource content from our users, encouraging them to share their own experiences with our products for a chance to get into our ‘PR and Influencers’ list. This approach created authentic brand stories, increased our engagement by 50%, and brought in a higher ROI than any of our previous campaigns.”

**58. How Do You Foster Innovation Within a Team?**

This question allows interviewers to examine your leadership style and how you cultivate a fertile environment for new ideas. The best answer will outline specific strategies you use to encourage your team to push boundaries and think creatively.

**Example of how to answer this interview question:**

- “I believe exposure to different perspectives sparks innovation. In my team, I facilitate cross-departmental workshops, which in multiple cases, resulted in projects that integrated unique insights from different teams, leading to innovative features in our products that set us apart from competitors.”

**59. Can You Provide an Example of a Project Where Your Creativity Made a Difference?**

Creativity in the workplace isn’t just about novelty. It’s about impactful innovation and real-world applications that drive progress and success. The best answer to this common job interview question illustrates your ability to think differently, use resources ingeniously, and bring in tangible results for your team or company.

**Example of how to answer this interview question:**

- “In my previous role as a **sales executive**, I was tasked with reviving a client’s interest in our services. Instead of the usual PowerPoint, I used an interactive web presentation tool to create a dynamic, engaging pitch. My creativity in presenting our solutions led to renewing a contract worth \$500K that was previously at risk.”

**60. What Is Your Process for Generating New Ideas?**

This interview question is a recruiter’s way of peering into your mental kitchen to see how you cook up recipes for innovation. A great answer should reveal your creative process, how you find inspiration, and the way you refine raw concepts into workable, practical solutions.

**Example of how to answer this interview question:**

- “My process starts with staying informed and dedicating time each week to reading industry publications and attending webinars. For instance, when I noticed a trending topic in

*cybersecurity, I proposed a new service offer for our IT company, which later became one of our most-in-demand services.”*

## 61. How Do You Stay Inspired and Keep Your Creative Skills Sharp?

Employers are always on the lookout for people who can keep their creativity fire burning long-term. With this question, they're trying to figure out if you have sustainable habits and strategies that fuel your imagination and innovation on a regular basis. Show them that instead of waiting for a muse to whisper in your ear, you go out there chasing.

**Example of how to answer this interview question:**

- *“To keep my creative juices flowing, I make it a point to step outside my comfort zone regularly. Last month, I joined a creative writing workshop, which not only honed my writing skills but also introduced me to new storytelling techniques that I translated into more engaging marketing content.”*

## 62. How Do You Encourage Others to Be Creative?

This question serves to identify potential leaders who understand that creativity isn't a solo performance but a group effort that can drive company success. Recruiters want to know whether you have the ability to encourage team members to think creatively and a strategic approach to nurturing such an important aspect.

**Example of how to answer this interview question:**

- *“I believe in giving team members ‘creative time’ to work on passion projects related to our business. At my last company, team members spent a few hours every Friday exploring new ideas. This approach led to the creation of two best-seller software products and many features that increased our customer base by 30%.”*

## 63. How Do You Evaluate the Feasibility of New Ideas?

This question is how interviewers try to understand if you possess both creativity and the practicality to implement ideas effectively. Your answer should communicate a methodical approach to assessment, demonstrating that you can balance creativity with analytical skills.

**Example of how to answer this interview question:**

- *“I believe in prototyping as a way to test the practicality of a new idea. I once led a team to develop a low-cost prototype of a new feature in our software, which we tested internally. This helped us identify potential issues and gather data on its usability before rolling it out for further development.”*

## 64. What Do You Do When Someone Challenges Your Ideas?

Being questioned or facing opposition to creative ideas is part of the collaborative process in any professional setting. Interviewers ask this question to learn more about your interpersonal skills,

flexibility, and how you handle critique.

**Example of how to answer this interview question:**

- *"I handle challenges to my ideas by preparing and presenting data that supports them. This was effective when my recommendation for a new product feature was initially met with skepticism but was later accepted after I shared user testing results that confirmed its potential. Once everyone was on board with my initial idea, we focused on refining it together."*

## Salary Interview Questions

This is the moment when the value you assign to your skills and experience meets what a potential employer is willing to offer. Turn a potentially uncomfortable conversation into an opportunity to show that you're worth every penny. Learn how to articulate your worth, [research salaries](#), and [negotiate like a pro](#) to ace these specific interview questions about salary.

### 65. What Are Your Salary Expectations?

Employers ask [questions about salary expectations](#) to ensure that your compensation requirements align with their budget and to prevent any miscommunication further down the hiring process. Your response should be well-researched and reflect your professional worth. It's a balancing act between asking for what you deserve and showing an understanding of the value you bring to the company.

**Example of how to answer this interview question:**

- *"In considering the responsibilities of this role and the industry standard, I expect a salary expectation between \$X and \$Y. However, I'm flexible and more interested in finding a position that is a good fit for my skills and ambitions, and I believe that the compensation will reflect that."*

### 66. Are You Willing to Negotiate Your Salary?

This question often serves to test your business acumen and your ability to handle sensitive discussions with poise and professionalism. The best answer to this question should show that you're both reasonable in your ask and confident in what you bring to the table.

**Example of how to answer this interview question:**

- *"It's important for me to receive fair compensation for my contributions to the company, and I understand that this is a two-way conversation. I'm ready to talk through the details to ensure that we're both happy with the arrangement."*

### 67. How Does Your Desired Salary Compare With the Industry Average for This Role?

This question reveals if you have a realistic expectation of how your experience and qualifications balance with the going rate for the job. Your answer should communicate that you know the industry standards and have considered where you fit.

**Example of how to answer this interview question:**

- *"In setting my desired salary, I have considered the industry average and how my track record of driving sales growth stands out. My expectation is competitive, considering my experience, the success I have achieved in similar roles, and the scope of responsibility for this position."*

## 68. What Benefits Beyond Salary Are Most Important to You?

Recruiters ask this question to learn what motivates you as a candidate beyond monetary compensation and if their benefits package aligns with your needs. Answer by providing a window into your priorities and values and explaining how a good benefits package can contribute to your work-life balance, personal growth, and job satisfaction.

**Example of how to answer this interview question:**

- *"Flexible working hours and the option for remote work are benefits that I prioritize highly. They allow for a better balance between my professional and personal life, enabling me to be more productive and satisfied in my role."*

## Technical Skills Interview Questions

These **competency-based interview questions** give you the chance to show off those valuable hard skills that make you a strong candidate. Employers want to be sure that your skill set matches the practical demands of the role. They're looking for concrete evidence that you're up to date with the latest tools, technologies, and methodologies in your field.

## 69. What Programming Languages Are You Proficient In?

This is an incredibly common interview question for roles that involve software development, data analysis, or any position requiring coding. The best answer will communicate your level of expertise with the programming languages you know, how they could be valuable to the employer, and any related special projects or accomplishments you've achieved.

**Example of how to answer this interview question:**

- *"I'm proficient in Java, Python, and C#. In my last role, I developed a suite of supply chain management tools in Java, automating processes that cut down on errors by 30%. I have also used Python for data analysis, providing insights that have driven strategic decisions, and I have developed several .NET applications with C#."*

## 70. Can You Explain the Steps You Take to Troubleshoot a Technical Problem?

This is the question recruiters ask to see if you have a systematic process for identifying, analyzing, and resolving issues, which is critical in any role that depends on technology. A solid answer should demonstrate your analytical abilities, attention to detail, and how you prioritize tasks. Show them you can handle pressure and uncertainty when things don't go as planned.

**Example of how to answer this interview question:**

- *"When troubleshooting, my first step is to replicate the issue to understand it better. For instance, when I receive reports about potential bugs in our application, I reproduce the problem in a test environment to isolate the cause. Then, I review the code for potential errors, consult the documentation, and if needed, discuss with colleagues to find further insights."*

## 71. Describe a Project Where You Used Data Analysis to Drive Decisions.

Data is king if you know how to interpret it and turn it into actionable strategies. Recruiters ask this question to assess your ability to harness the power of data to solve problems, predict trends, and advise business decisions. When answering, clarify the methods you use to analyze data, the tools and technologies you rely on, and the impact of your data-driven decisions.

**Example of how to answer this interview question:**

- *"While working on improving user experience for our e-commerce platform, I used A/B testing and heat map data to analyze user interaction with our site. My analysis led to a redesign of the site's navigation, which ultimately increased our conversion rate by 30% and significantly lowered the bounce rate."*

## 72. How Do You Ensure the Quality of Your Work When Performing Technical Tasks?

In technical work, quality assurance is a fundamental aspect that runs parallel with every task you undertake. This question assesses your commitment to excellence and your proactive measures in maintaining high standards. Your answer should reflect a meticulous and systematic approach to your work.

**Example of how to answer this interview question:**

- *"When writing or reviewing critical code, I ensure quality by maintaining comprehensive documentation and change logs. In my previous role, this helped streamline the review process, facilitated troubleshooting post-deployment, and consequently reduced downtime for our clients."*

## 73. Tell Me About a Time You Learned a New Technology to Complete a Job Task.

Interviewers ask this question to understand how you adapt to technological advancements and your commitment to continuous learning in a rapidly evolving workplace. They want to evaluate

your ability to take initiative and self-teach, and how you apply new knowledge to enhance job performance. Your answer should reflect your proactive approach to professional growth and adaptability.

**Example of how to answer this interview question:**

- *"In my last role, I had the task of optimizing our website's SEO. Although I had basic knowledge, I completed an advanced SEO strategy course and applied these new techniques to our website. Within three months, we saw a 40% increase in traffic and a significant boost in our page ranking for key search terms."*

## 74. What Software Tools Do You Have Experience With?

Share with your interviewers the tools relevant to your field that you've used and your proficiency level with each. It's also helpful to mention how you've used these tools to achieve results in your previous role, illustrating practical use and impact.

**Example of how to answer this interview question:**

- *"I have extensive experience with Adobe Creative Suite, particularly Photoshop and InDesign. At my previous job, I used these tools to design marketing materials that contributed to a 30% increase in engagement on our social media platforms."*

## 75. Can You Describe Your Experience with Automation or Improving Efficiency Through Technology?

In a world where efficiency is synonymous with a competitive edge, companies are perpetually hunting for ways to do more with less effort and time. A strong answer to this question demonstrates your initiative in identifying opportunities for automation, your problem-solving skills in implementing these solutions, and the outcomes of your efforts.

**Example of how to answer this interview question:**

- *"During my tenure as a customer service manager, I introduced chatbots to handle our initial customer inquiries. This freed up our customer service reps to deal with more complex cases, increasing our resolution rate for advanced issues by 50% and overall customer satisfaction by 20%."*

## More Technical Skills Interview Questions

Turn your expertise into job offers. For more [technical interview questions to prepare for](#), check out the following articles:

- [SQL interview questions](#)
- [Java interview questions](#)
- [JavaScript interview questions](#)

Here's how to answer the 16 most common interview questions:

- 1.** Tell me about yourself.
- 2.** How would you describe yourself?
- 3.** What makes you unique?
- 4.** Why do you want to work here?
- 5.** What interests you about this role?
- 6.** What motivates you?
- 7.** What are you passionate about?
- 8.** Why are you leaving your current job?
- 9.** What are your greatest strengths?
- 10.** What are your greatest weaknesses?
- 11.** What are your goals for the future?
- 12.** Where do you see yourself in five years?
- 13.** Can you tell me about a difficult work situation and how you overcame it?
- 14.** What is your salary range expectation?
- 15.** Why should we hire you?
- 16.** Do you have any questions?

Every interviewer is different and each one's questions may vary. By preparing answers for these common interview questions, you can develop compelling talking points to make a great impression during your next job interview.

## Related jobs on Indeed

[\*\*Part-time jobs\*\*](#) >[\*\*Full-time jobs\*\*](#) >[\*\*Remote jobs\*\*](#) >[\*\*Urgently needed jobs\*\*](#) >[\*\*View more jobs on Indeed\*\*](#) →

## Best interview questions and answers

### 1. Tell me about yourself.

Your interviewers will likely start out with a question about you and your background to get to know you. Start by giving them an overview about your current position or activities then provide the most important and relevant highlights from your background that make you the most qualified for the role. If you'd like, it is generally acceptable to include some light personal details about things like your pets, hobbies or family. Doing so can help you be more memorable and personable to the interviewer.

**Example:** *"Currently, I serve as the assistant to three of the company's five executive team members including the CEO. During my time at the organisation, I have been recognised for my time management skills, writing abilities and commitment to excellence."*

From my 12 years of experience as an executive assistant, I've developed the ability to anticipate roadblocks and create effective alternative plans. My greatest value to any executive is my ability to work independently, freeing up their time to focus on the needs of the business.

It's clear that you're looking for someone who understands the nuances of managing a CEO's busy day and can proactively tackle issues. As someone with a sharp eye for detail and a drive to organise, I thrive on making sure every day has a clear plan and every plan is clearly communicated."

For more on answering this question, visit [Interview Question: "Tell Me About Yourself."](#) ↗

### 2. How would you describe yourself?

When interviewers ask you to talk about yourself, they're looking for information about how your qualities and characteristics align with the skills they believe is required to succeed in the role. If possible, include quantifiable results to demonstrate how you use your best attributes to drive success.

**Example:** *"I am a vigilant and proactive Security Officer working to ensure safe, secure and orderly environments. I'm also a lifelong learner always seeking out the latest security equipment and techniques to patrol buildings. Lastly, I am thorough in documenting all incidents and actively making suggestions to management about security improvements and changes."*

For more on answering this question, visit [Interview Question: "How Would You Describe Yourself?"](#)

### 3. What makes you unique?

Employers often ask this question to identify why you might be more qualified than other candidates they're interviewing. To answer, focus on why hiring you would benefit the employer. As you don't know the other applicants, it can be challenging to think about your answer in relation to theirs. Addressing why your background makes you a good fit will let employers know why your traits and qualifications make you well prepared.

**Example:** *"What makes me unique is my experience of four years in retail. Because I've had first-hand experience fielding shoppers' questions, feedback and complaints, I know what customers want. I know what it takes to create a positive consumer experience through marketing."*

### 4. Why do you want to work here?

Interviewers often ask this question as a way to determine whether or not you took time to research the company and to learn why you see yourself as a good fit. The best way to prepare for this question is to do your homework and learn about the products, services, mission, history and culture of this workplace. In your answer, mention the aspects of the company that appeal to you and align with your career goals. Explain why you're looking for these things in an employer.

**Example:** *"The company's mission to help college graduates pay off their student loan debt speaks to me. I've been in that situation and I'd love the opportunity to work with a company that's making a difference. Finding a company with a positive work environment and values that align with my own has remained a priority throughout my job search and this company ranks at the top of the list."*

## 5. What interests you about this role?

Like the previous question, hiring managers often include this question to make sure you understand the role and to give you the opportunity to highlight your relevant skills. In addition to thoroughly reading the job description, it can be helpful to compare the role requirements against your skills and experience. Choose a few things you particularly enjoy or excel at and focus on those in your answer.

**Example:** *"I've been passionate about user experience design for most of my professional career. I was excited to see this company uses Adobe products because I'm well versed in the entire suite. Also, I'm a huge advocate for applying agile workflows to design. I think it's the most effective way to tackle large projects. I was able to successfully build and launch an agile process in my previous role as UX manager and we saw considerable improvements in project speed."*

## 6. What motivates you?

Employers ask this question to gauge your level of self-awareness and ensure your sources of motivation align with the role. To answer, be as specific as possible, provide real-life examples and tie your answer back to the job role.

**Example:** *"Making a true difference in the lives of my patients and their families motivates me to strive for excellence in everything I do. I look forward to seeing my patients' reaction when we get a positive outcome that will change their lives forever. That's why I became a nurse and why I'm pursuing a position in pediatrics."*

## 7. What are you passionate about?

Much like the previous question about motivation, employers might ask what you are passionate about to better understand what drives you and what you care most deeply about. This can help them understand whether you are a good fit for the role and if it fits into your larger goals. To answer, select something you are genuinely passionate about, explain why you're passionate about it, give examples of how you've pursued this passion and relate it back to the job.

**Example:** *"As an experienced, service-oriented professional with more than a decade of experience working in boutique salons, I thrive on creating a welcoming environment for all clients and providing the highest quality skincare services. My specialised training, along with my interpersonal skills, has helped me become adept at developing long-term, trusted relationships that help to build a loyal client base. These relationships are the reason I'm excited to go to work every day."*

## 8. Why are you leaving your current job?

There are many reasons for leaving a job. Prepare a thoughtful answer that will give your interviewer confidence that you're being deliberate about this job change. Instead of focusing on the negative aspects of your current or previous role, focus on the future and what you hope to gain in your next position.

**Example:** *"I'm looking for an opportunity that gives me the ability to build closer, long-term relationships with clients. In my current role, the sales cycle is so short that I don't spend as much time building a rapport with my customers as I'd like. Relationship-building is one of the reasons I chose a career in sales and I look forward to working with a company in which that's a top priority."*

## 9. What are your greatest strengths?

This question gives you an opportunity to talk about your technical and soft skills. To answer, share qualities and personal attributes and then relate them back to the role for which you're interviewing.

**Example:** *"I'm a natural problem solver. I find it rewarding to dig deep and uncover solutions to challenges. It's like solving a puzzle. It's something I've always excelled at and something I enjoy. Much of product development is about finding innovative solutions to challenging issues, which is what drew me to this career path in the first place."*

For more on answering this question, visit [Interview Question: "What Are Your Strengths And Weaknesses?"](#)

## 10. What are your greatest weaknesses?

It can feel awkward to discuss your weaknesses in an environment where you're expected to focus on your accomplishments. However, when answered correctly, sharing your weaknesses can show that you are self-aware and want to continuously get better at your job—traits that are extremely attractive to many employers. Remember to start with the weakness and then discuss the measures you've taken to improve. This way, you're finishing your answer on a positive note.

**Example:** *"I sometimes have trouble saying 'no' and end up overwhelmed by my workload. Earlier in my career, I would take on so many projects that I'd work evenings and weekends. It was stressful. I realised this was counterproductive so I started using workload management tools and setting better expectations for myself and my teammates."*

## 11. What are your goals for the future?

Often, hiring managers ask about your future goals to determine whether or not you're looking to stay with the company in the long term. Additionally, this question is used to gauge your ambition, expectations for your career and your ability to plan ahead. The best way to handle this question is to determine your current career trajectory and how this role plays into helping you reach your ultimate goals.

**Example:** *"I would like to continue developing my marketing expertise as well as my leadership skills over the next several years. One of the reasons I'm interested in working for a fast-growing start-up company is that I'll have the ability to wear many hats and collaborate with many different departments. I believe this experience will serve me well in achieving my ultimate goal of leading a marketing department someday."*

## 12. Where do you see yourself in five years?

Understanding how you imagine your life in the future can help employers understand whether the trajectory of the role and company fits in with your personal development goals. To answer, provide general ideas about the skills you want to develop, the types of roles you would like to be in and the things you would like to have accomplished.

**Example:** *"In five years, I'd like to be an industry expert in my field with the ability to train and mentor students and entry-level designers. I would also like to gain specialised knowledge in user experience to be a well-rounded contributor working with design and marketing teams on large scale projects that make a difference in the company and to the global community."*

## 13. Can you tell me about a difficult work situation and how you overcame it?

This question is often used to assess how well you perform under pressure as well as your problem-solving abilities. Keep in mind stories are more memorable than facts and figures so strive to 'show' instead of 'tell'. This is also an excellent opportunity to show your human side and how you're willing to go the extra mile without being asked.

**Example:** *"It was the first day of my boss's two-week vacation and our agency's highest-paying client threatened to leave because he didn't feel he was getting the personalised service he was promised. I spent my lunch hour on the phone with him talking through his concerns. We even brainstormed ideas for his next campaign. He was so grateful for the personal attention that he signed another six-month contract before my boss even returned from her trip."*

## 14. What is your salary range expectation?

Interviewers ask this question to make sure your expectations are in line with the amount they've budgeted for the role. If you give a salary range exceedingly lower or higher than

the market value of the position, it gives the impression that you don't know your worth. Research the typical compensation range for the role on [Indeed Salaries](#) and request for the higher side of your range. Be sure to let the hiring manager know if you're flexible with your rate.

**Example:** *"My salary expectation is between INR XX,XXX and INR XX,XXX, which is the average salary for a candidate with my level of experience in this city. However, I am flexible."*

## 15. Why should we hire you?

While this question may seem like an intimidation tactic, interviewers generally bring this up to offer you another opportunity to explain why you're the best candidate. Your answer should address the skills and experience you offer and why you're a good culture fit.

**Example:** *"I have a passion for application development that's grown stronger over the course of my five-year career. The company's mission aligns with my personal values and from my limited time in the office, I can already tell this is the sort of positive culture in which I would thrive. I want to work for a company that has the potential to reshape the industry and I believe you're doing just that."*

For more on answering this question, visit [Interview Question: "Why Should We Hire You?"](#)

## 16. Do you have any questions?

This might be one of the most important questions asked during the interview process because it allows you to explore any subject that hasn't been addressed and shows the interviewer you're excited about the role. By this point, you'll likely have already covered most of the basics about the position and the company so take time to ask the interviewer questions about their own experiences with the company and gain tips on how you can succeed if hired.

**Example:** *"What do you love about working for this company? What would success look like in this role? What are some of the challenges people typically face in this position?"*

Much like preparing for a test in school, the best way to succeed in your interview is to study and practice. Do research on the company and the job and practice your talking points until you feel confident about your answers. The more you prepare, the more likely you are to leave a lasting impression and outperform fellow candidates.

### Related:

- [5 Work-Life Balance Interview Questions \(With Answers\)](#)



Where your work meets your life.

**Harvard  
Business  
Review**

**Career Planning**

# 10 Common Job Interview Questions and How to Answer Them

by Vicky Oliver

November 11, 2021



HBR Staff/Getty Images/svetolk

**Summary.** Interviews can be high stress, anxiety-driving situations, especially if it's your first interview. A little practice and preparation always pays off. While we can't know exactly what an employer will ask, here are 10 common interview questions along with advice on how to... [more](#)



Where your work meets your life. See more from Ascend here.

---

Resignation numbers have remained abnormally high in the U.S. between July 2021 and October 2021, with millions of Americans quitting their jobs — which also means there are millions of new openings up for grabs. If you're entering the market for the first time, or just looking to make a change, use this guide to prepare for your next interview.

Below is a list of 10 common job interview questions, along with answering techniques that will help you dazzle your prospects, and hopefully, secure the role you want.

### **1. Could you tell me about yourself and describe your background in brief?**

Interviewers like to hear stories about candidates. Make sure your story has a great beginning, a riveting middle, and an end that makes the interviewer root for you to win the job.

Talk about a relevant incident that made you keen on the profession you are pursuing and follow up by discussing your education. In the story, weave together how your academic training and your passion for the subject or industry the company specializes in, combined with your work experience, make you a great fit for the job. If you've managed a complex project or worked on an exciting, offbeat design, mention it.

**Example:** “I come from a small town, where opportunities were limited. Since good schools were a rarity, I started using online learning to stay up to date with the best. That’s where I learned to code and then I went on to get my certification as a computer programmer. After I got my first job as a front-end coder, I continued to invest time in mastering both front- and back-end languages, tools, and frameworks.”

## **2. How did you hear about this position?**

Employers want to know whether you are actively seeking out their company, heard of the role from a recruiter, or were recommended to the position by a current employee. In short, they want to know how you got to them.

If someone recommended you for the position, be sure to say their name. Don't assume that the interviewer already knows about the referral. You'll probably want to also follow up with how you know the person who referred you. For example, if you and Steve (who recommended you) worked together previously, or if you met him over coffee at a networking event, mention it to give yourself a little more credibility. If Steve works at the company and suggested that you apply for the job, explain why he thought you'd be the perfect fit.

If you sought out the role yourself, be clear about what caught your eye — extra bonus points if you can align your values with the company and their mission. You want to convince the hiring manager that you chose their company, over all other companies, for a few specific reasons.

Lastly, if you were recruited, explain why you took the bait. Did this role sound like a good fit? Does it align with the direction you want to take your career? Even if you weren't familiar with the organization prior to being recruited, be enthusiastic about what you've learned and honest about why you're interested in moving forward with the process.

**Example:** "I learned about the position through LinkedIn as I've been following your company's page for a while now. I'm really passionate about the work you're doing in X, Y, and Z areas, so I was excited to apply. The required skills match well with the skills I have, and it seems like a great opportunity for me to contribute to your mission, as well as a great next move for my career."

### **3. What type of work environment do you prefer?**

Be sure to do your homework on the organization and its culture before the interview. Your research will save you here. Your preferred environment should closely align to the company's workplace culture (and if it doesn't, it may not be the right fit for you). For example, you may find on the company's website that they have a flat organizational structure or that they prioritize collaboration and autonomy. Those are key words you can mention in your answer to this question.

If the interviewer tells you something about the company that you didn't uncover in your research, like, "Our culture appears buttoned-up from the outside, but in reality, it's a really laid-back community with little competition among employees," try to describe an experience you've had that dovetails with that. Your goal is to share how your work ethic matches that of the organization's.

**Example:** "That sounds great to me. I like fast-paced work environments because they make me feel like I'm always learning and growing, but I really thrive when I'm collaborating with team members and helping people reach a collective goal as opposed to competing. My last internship was at an organization with a similar culture, and I really enjoyed that balance."

### **4. How do you deal with pressure or stressful situations?**

The employer wants to know: Do you hold down the fort or crumble under pressure? They want to make sure that you won't have a meltdown when the pressure becomes intense and deadlines are looming. The ability to stay calm under pressure is a highly prized talent.

Share an instance when you remained calm despite the turmoil. If it's a skill you're developing, acknowledge that and include the steps you're taking to respond better to pressure in the future. For example, you could indicate that you've started a mindfulness practice to help you better deal with stress.

**Example:** “I realize stressful situations are always going to come up, and I definitely have had to learn how to navigate them throughout my career. I think I get better at it with every new experience. While working on a new product launch at my last company, for example, things were not going according to plan with my team. Instead of pointing fingers, my first reaction was to take a step back and figure out some strategies around how we could solve the problem at hand. Previously, I may have defaulted to panicking in that situation, so being calm and collected was definitely a step forward and helped me approach the situation with more clarity.”

## **5. Do you prefer working independently or on a team?**

Your answer should be informed by the research you’ve done on the company culture and the job in question. Nevertheless, you should expect that most work environments will have some team aspect.

Many positions require you to work collaboratively with other people on a daily basis, while some roles require you to work on your own. When you answer this question, highlight the best traits of your personality and how they fit the job requirements. It could also be in your interest to answer this question by highlighting the advantages and disadvantages of both situations.

**Example:** “I enjoy a blend of the two. I like having a team to strategize with, get diverse opinions from, and reach out to for feedback. But I am also comfortable taking on assignments that require me to work independently. I find I do some of my best work when I can focus alone in a quiet space, but I really value collaborating with my teammates to come up with the best ideas.”

## **6. When you’re balancing multiple projects, how do you keep yourself organized?**

Employers want to understand how you use your time and energy to stay productive and efficient. They’re also looking to understand if you have your own system for staying on track with

the work beyond the company's schedules and workflow plans. Be sure to emphasize that you adhere to deadlines and take them seriously.

Discuss a specific instance when you stayed on track. Talk about the importance and urgency of the projects you were working on and how you allocated your time accordingly. Explain how you remain organized and focused on the job in front of you.

**Example:** "I'm used to juggling projects at my current job where I'm often moving between coding one software program to another. I use the timeboxing technique to make sure they're all on track, allocating time on my calendar for certain tasks. I've found it really helps me prioritize what needs to get done first, and it holds me accountable for the more repetitive day-to-day tasks I'm responsible for."

## **7. What did you do in the last year to improve your knowledge?**

This question may come up as a result of the pandemic. Employers want to know how people used their time differently. Know that you don't have to feel scared about answering this question if you didn't spend your time brushing up on skills or taking courses. We learn from any experience we have.

If you spent time honing your professional skills, you might say the following.

**Example:** "The extra time on my plate really allowed me to get introspective around where I want to take my career. I read a lot of journals to keep abreast of the latest ideas in my field, and sharpened my skills by taking some online courses, such as..." (and then be specific).

If you chose to work on your personal development, you could say something like the following.

**Example:** “Like everyone else, I, too, gained some time last year from not having to travel two hours a day to and from work. I decided to spend my time on things I love. So I got back to learning how to play the guitar and journaling. I feel it brought me closer to myself and has been really great for my mental health and productivity.”

## **8. What are your salary expectations?**

Before you walk in for your first interview, you should already know what the salary is for the position you’re applying to. Check out websites such as Glassdoor, Fishbowl, or Vault.com for salary information. You could also ask people in the field by reaching out to your community on LinkedIn.

Employers will always ask this question because every position is budgeted, and they want to ensure your expectations are consistent with that budget before moving forward.

Remember that it’s often better to discuss a salary range rather than a specific number during the interview and leaving room for negotiation. It’s also better to err on the side of caution and quote a slightly higher number as it’s easier to negotiate downward than upward. As a general rule of thumb, I advise not bringing up the questions about salary until your interviewer does or bringing it up too early in the process.

**Example:** “Based on my skills and experience and on the current industry rates, I’m looking at a salary around \$\_\_\_\_” (then fill in with your desired salary range and rationale).

## **9. Are you applying for other jobs?**

Interviewers want to know if you’re genuinely interested in this position or if it’s just one of your many options. Simply, they want to know if you’re their top choice. Honesty is the best policy. If you’re applying for other jobs, say so. You don’t have to necessarily say where you’re applying unless you have another

offer. But they might want to know where in the hiring process you are with other companies. You can also mention that you're actively looking for offers if your interviewer asks.

**Example:** "I've applied to a couple of other firms, but this role is really the one I'm most excited about right now because..."

## **10. From your resume it seems you took a gap year. Would you like to tell us why that was?**

Gap years are more popular in some cultures than others. In some professions, gap years may have a negative connotation (the industry moves too fast and you're not up to date).

Let your interviewer know that your gap year wasn't about procrastinating over your transition from childhood to adulthood, but that it added value to the confident professional you have become. Based on what part of the world you're in and how common these are, employers are likely looking to hear stories of what you did and how your experiences have benefitted and prepared you for this role.

Provide a short explanation of why you decided to pursue a gap year, then focus on what came out of it that made a positive difference for your future.

**Example:** "During my last year of high school, I didn't feel ready to choose my educational path, so I took a wilderness course for a few months to sort out my life goals. It may seem a little random, but the time I spent actually helped me develop so many new skills — in the areas of leadership, communication, (etc...). During that time, I realized that I wanted to earn a degree in (state your degree) to align with my passion (say what that is)."

To make a winning impression, you'll need to answer each question with poise and passion. But practicing first really helps. Meticulous preparation will allow you to appear confident and in control, helping position you as the ideal candidate when the competition is tough.

**Vicky Oliver** is a leading career development expert and the multi-best-selling author of five books, including ***301 Smart Answers to Tough Interview Questions***, named in the top 10 list of “Best Books for HR Interview Prep.” She’s a sought-after speaker and seminar presenter and a popular media source, having made over 900 appearances in broadcast, print, and online outlets.

## Recommended For You

---

### Good (and Bad) Answers to Common Interview Questions



---

#### PODCAST

#### How Do I Avoid a Career Plateau at Midlife?



---

### 5 Ways to Prepare for Situational Interview Questions



---

### How to Show You're a "Good Fit" During an Interview



Search jobs or companies[ADVICE](#) / [JOB SEARCH](#) / [INTERVIEWING](#)

# 50+ Most Common Interview Questions and Answers

by [The Muse Editors](#)

Updated 3/29/2024



*Getty Images*

Wouldn't it be great if you knew exactly what questions a [hiring manager](#) would be asking you in your next [job interview](#)?

We can't read minds, unfortunately, but we'll give you the next best thing: a list of 53 of the most commonly asked interview questions and answers, along with advice on how to come up with your own responses.

While we don't recommend having a canned response for every interview question (in fact, please don't), we *do* recommend spending some time getting comfortable with what you might be asked, what hiring managers are really looking for in your responses, and what it takes to show that you're the right person for the job.

## ***Land more interviews by looking for [more open jobs](#) on The Muse »***

Consider this list your job interview answer and question study guide. (And don't miss our bonus list at the end, with links to resources on specific types of interview questions—about emotional intelligence or [diversity and inclusion](#), for example—and interview questions by role, from accountant to project manager to teacher.)

# **50+ most common job interview questions and answers**

- [Tell me about yourself.](#)
- [Walk me through your resume.](#)
- [How did you hear about this position?](#)
- [Why do you want to work at this company?](#)
- [Why do you want this job?](#)
- [Why should we hire you?](#)
- [What can you bring to the company?](#)
- [What are your greatest strengths?](#)
- [What do you consider to be your weaknesses?](#)
- [What is your greatest professional achievement?](#)
- [Tell me about a challenge or conflict you've faced at work, and how you dealt with it.](#)
- [Tell me about a time you demonstrated leadership skills.](#)
- [What's a time you disagreed with a decision that was made at work?](#)
- [Tell me about a time you made a mistake.](#)
- [Tell me about a time you failed.](#)
- [Why are you leaving your current job?](#)
- [Why were you fired?](#)
- [Why was there a gap in your employment?](#)

- [Can you explain why you changed career paths?](#)
- [What's your current salary?](#)
- [What do you like least about your job?](#)
- [What are you looking for in a new position?](#)
- [What type of work environment do you prefer?](#)
- [What's your work style?](#)
- [What's your management style?](#)
- [How would your boss and coworkers describe you?](#)
- [How do you deal with pressure or stressful situations?](#)
- [What do you like to do outside of work?](#)
- [Are you planning on having children?](#)
- [How do you stay organized?](#)
- [How do you prioritize your work?](#)
- [What are you passionate about?](#)
- [What motivates you?](#)
- [What are your pet peeves?](#)
- [How do you like to be managed?](#)
- [Do you consider yourself successful?](#)
- [Where do you see yourself in five years?](#)
- [How do you plan to achieve your career goals?](#)
- [What are your career aspirations?](#)
- [What's your dream job?](#)
- [What other companies are you interviewing with?](#)
- [What makes you unique?](#)
- [What should I know that's not on your resume?](#)

- [What would your first few months look like in this role?](#)
- [What are your salary expectations?](#)
- [What do you think we could do better or differently?](#)
- [When can you start?](#)
- [Are you willing to relocate?](#)
- [How many tennis balls can you fit into a limousine?](#)
- [If you were an animal, which one would you want to be?](#)
- [Sell me this pen.](#)
- [Is there anything else you'd like us to know?](#)
- [Do you have any questions for us?](#)

## 1. Tell me about yourself.

This question seems simple, so many people fail to prepare for it, but it's crucial. Here's the deal: Don't give your complete employment (or personal) history. Instead, give a pitch—one that's concise and compelling and that shows exactly why you're the right fit for the job.

Muse writer and MIT career counselor Lily Zhang recommends using a present, past, future formula. Talk a little bit about your current role (including the scope and perhaps one big accomplishment), then give some background as to how you got there and experience you have that's relevant. Finally, segue into why you want—and would be perfect for—this role.

### Possible answer to “Tell me about yourself.”

*“Well, I’m currently an account executive at Smith, where I handle our top-performing client. Before that, I worked at an agency where I was on three different major national healthcare brands. And while I really enjoyed the work that I did, I’d love the chance to dig in much deeper with one specific healthcare company, which is why I’m so excited about this opportunity with Metro Health Center.”*

**Read More:** [A Complete Guide to Answering “Tell Me About Yourself” in an Interview \(Plus Examples!\)](#)

## 2. Walk me through your resume.

Like “Tell me about yourself,” this question is a common interview opener. But instead of framing your answer around what qualities and skills make you best for the position, your answer should group your qualifications by your past jobs and tell your career story. You might choose to tell this story chronologically, especially if there’s a great anecdote about what set you on this path. Or, as with “Tell me about yourself,” you can begin with your present job then talk about what brought you here and where you’re going next. But regardless, when you speak about your “past” and “present,” highlight your most relevant experiences and accomplishments for *this* job and wrap up by talking about the future, i.e. connect your past and present together to show why this job should be the next one you add to your resume.

### Possible answer to “Walk me through your resume.”

*“Well, as you can see from my resume, I took a bit of a winding road to get to where I am today. In college, I double majored in chemistry and communications. I found early on that working in a lab all day wasn’t for me and at some point I realized I looked forward to the lab class I TA’ed the most.*

*“So when I graduated, I found a job in sales for a consumer healthcare products company, where I drew on my teaching experience and learned even more about tailoring your message and explaining complex health concepts to people without a science background. Then, I moved into a sales training role at a massive company where I was responsible for teaching recent graduates the basics of selling. My trainees on average had more deals closed in their first quarter than any of the other trainers’ cohorts. Plus, I got so much satisfaction from finding the right way to train each new hire and watching them progress and succeed. It reminded me of my time as a TA in college. That’s when I started taking night classes to earn my chemistry teaching certificate.*

*“I left my full-time job last year to complete my student teaching at P.S. 118 in Manhattan, and over the summer, I worked for a science camp, teaching kids from the ages of 10 to 12 about basic chemistry concepts and best practices for safe experiments. Now, I’m excited to find my first full-time teaching job, and your district is my top choice. The low student-to-teacher ratio will let me take the time to teach each student in the best way for them—which is my favorite part of the job.”*

**Read More:** [How to Respond to “Walk Me Through Your Resume”—and Get Your Interview Started on the Right Note](#)

## 3. How did you hear about this position?

Another seemingly innocuous interview question, this is actually a perfect opportunity to stand out and show your passion for and connection to the company. For example, if you found out about the gig through a friend or professional contact, name-drop that person, then share why you were so excited about the job. If you discovered the company through an event or article, share that. Even if you found the listing through a random job board, share what, specifically, caught your eye about the role.

### Possible answer to “How did you hear about this position?”

*“I heard about an opening on the product team through a friend of a friend, Akiko, and since I’m a big fan of your work and have been following you for a while I decided it would be a great role for me to apply for.”*

**Read More:** [3 Ways People Mess Up the \(Simple\) Answer to “How Did You Come Across This Job Opportunity?”](#)

## 4. Why do you want to work at this company?

Beware of generic answers! If what you say can apply to a whole slew of other companies, or if your response makes you sound like every other candidate, you’re missing an opportunity to stand out. Zhang recommends one of four strategies: Do your research and point to something that makes the company unique that really appeals to you; talk about how you’ve watched the company grow and change since you first heard of it; focus on the organization’s opportunities for future growth and how you can contribute to it; or share what’s gotten you excited from your interactions with employees so far. Whichever route you choose, make sure to be specific. And if you can’t figure out why you’d want to work at the company you’re interviewing with by the time you’re well into the hiring process? It might be a red flag telling you that this position is not the right fit.

### Possible answer to “Why do you want to work at this company?”

*"I saw on The Muse that you were also hiring for new positions on the West Coast to support your new operations there. I did some more reading about the new data center you're building there and that excites me as I know this means there'll be opportunities to train new teammates. I also learned through a Wall Street Journal article that you're expanding in Mexico as well. I speak Spanish fluently and would be eager to step up and help liaise whenever necessary."*

**Read More:** [4 Better Ways to Answer “Why Do You Want to Work at This Company?”](#)

## 5. Why do you want this job?

Again, companies want to hire people who are passionate about the job, so you should have a great answer about why you want the position. (And if you don't? You probably should apply elsewhere.) First, identify a couple of key factors that make the role a great fit for you (e.g., “I love customer support because I love the constant human interaction and the satisfaction that comes from helping someone solve a problem”), then share why you love the company (e.g., “I've always been passionate about education, and I think you're doing great things, so I want to be a part of it”).

### Possible answer to “Why do you want this job?”

*“I've always been a fan of X Co's products and I've spent countless hours playing your games. I know that your focus on unique stories is what drew me and other fans into your games initially and keeps us coming back for more. I've followed X Co on social media for a while, and I've always loved how you have people in different departments interact with users. So I was psyched when I came across this posting for a social media manager with TikTok experience. At my last job, I was responsible for launching our TikTok account and growing it to 10,000 followers in six months. Between that experience, my love of gaming, and my deep knowledge of your games and fanbase, I know I could make this TikTok account something special and exciting.”*

**Read More:** [3 Steps for Answering “Why Do You Want This Job?”](#)

## 6. Why should we hire you?

This interview question seems forward (not to mention intimidating!), but if you're asked it, you're in luck: There's no better setup for you to sell yourself and your skills to the hiring

manager. Your job here is to craft an answer that covers three things: that you can not only do the work, but also deliver great results; that you'll really fit in with the team and culture; and that you'd be a better hire than any of the other candidates.

## Possible answer to “Why should we hire you?”

*“I know it’s been an exciting time for General Tech—growing so much and acquiring several startups—but I also know from experience that it can be challenging for the sales team to understand how new products fit in with the existing ones. It’s always easier to sell the product you know, so the newer stuff can get shortchanged, which can have company-wide ramifications. I have over a decade of experience as a sales trainer, but more importantly, most of those years were working with sales teams that were in the exact same boat Gen Tech is in now. Growth is wonderful, but only if the rest of the company can keep up. I’m confident I can make sure your sales team is confident and enthusiastic about selling new products by implementing an ongoing sales training curriculum that emphasizes where they sit in a product lineup.”*

**Read More:** [3 Better Ways to Answer “Why Should We Hire You?”](#)

## 7. What can you bring to the company?

When interviewers ask this question, they don’t just want to hear about your background. They want to see that you understand what problems and challenges they’re facing as a company or department as well as how you’ll fit into the existing organization. Read the job description closely, do your research on the company, and make sure you pay attention in your early round interviews to understand any issues you’re being hired to solve. Then, the key is to connect your skills and experiences to what the company needs and share an example that shows how you’ve done similar or transferable work in the past.

## Possible answer to “What can you bring to the company?”

*“As Jocelyn talked about in our interview earlier, PopCo is looking to expand its market to small business owners with less than 25 employees, so I’d bring my expertise in this area and my experience in guiding a sales team that’s selling to these customers for the first time. In most of my past roles, this segment has been my focus and in my current role, I also played a big part in creating our sales strategies when the business began selling to these customers. I worked with my managers to develop the sales script. I also listened in on a number of sales calls with other account execs who were selling to these customers*

*for the first time and gave them pointers and other feedback. In the first quarter, our 10-person sales team closed 50 new bookings in this segment, and I personally closed 10 of those deals. I helped guide my last company through the expansion into small businesses, and I'm eager to do that again at PopCo. Plus, I noticed you have a monthly karaoke night—so I'm eager to bring my rendition of 'Call Me Maybe' to the team as well."*

**Read More:** [What Interviewers Really Want to Hear When They Ask "What Can You Bring to the Company?"](#)

## 8. What are your greatest strengths?

Here's an opening to talk about something that makes you great—and a great fit for this role. When you're answering this question, think quality, not quantity. In other words, don't rattle off a list of adjectives. Instead, pick one or a few (depending on the question) specific qualities that are relevant to this position and illustrate them with examples. Stories are always more memorable than generalizations. And if there's something you were hoping to mention because it makes you a great candidate, but you haven't had a chance yet, this would be the perfect time.

### Possible answer to "What are your greatest strengths?"

*"I'd say one of my greatest strengths is bringing organization to hectic environments and implementing processes to make everyone's lives easier. In my current role as an executive assistant to a CEO, I created new processes for pretty much everything, from scheduling meetings to planning monthly all hands agendas to preparing for event appearances. Everyone in the company knew how things worked and how long they would take, and the structures helped alleviate stress and set expectations on all sides. I'd be excited to bring that same approach to an operations manager role at a startup, where everything is new and constantly growing and could use just the right amount of structure to keep things running smoothly."*

**Read More:** [3 Smart Strategies for Answering "What's Your Greatest Strength?"](#)

## 9. What do you consider to be your weaknesses?

What your interviewer is really trying to do with this question—beyond identifying any major red flags—is to gauge your self-awareness and honesty. So, "I can't meet a deadline to

“Save my life” is not an option—but neither is “Nothing! I’m perfect!” Strike a balance by thinking of something that you struggle with but that you’re working to improve. For example, maybe you’ve never been strong at public speaking, but you’ve recently volunteered to run meetings to help you get more comfortable when addressing a crowd.

## Possible answer to “What do you consider to be your weaknesses?”

*“It can be difficult for me to gauge when the people I’m working with are overwhelmed or dissatisfied with their workloads. To ensure that I’m not asking too much or too little from my team, we have weekly check-ins. I like to ask if they feel like they’re on top of their workload, how I could better support them, whether there’s anything they’d like to take on or get rid of, and if they’re engaged by what they’re doing. Even if the answer is ‘all good,’ these meetings really lay the groundwork for a good and trusting relationship.”*

**Read More:** [4 Ways to Answer “What Is Your Greatest Weakness?” That Actually Sound Believable](#)

## 10. What is your greatest professional achievement?

Nothing says “hire me” better than a track record of achieving amazing results in past jobs, so don’t be shy when answering this interview question! A great way to do so is by using the [STAR method](#): situation, task, action, results. Set up the situation and the task that you were required to complete to provide the interviewer with background context (e.g., “In my last job as a junior analyst, it was my role to manage the invoicing process”), then describe what you did (the action) and what you achieved (the result): “In one month, I streamlined the process, which saved my group 10 person-hours each month and reduced errors on invoices by 25%.”

## Possible answer to “What is your greatest professional achievement?”

*“My greatest accomplishment was when I helped the street lighting company I worked for convince the small town of Bend, Oregon to convert antiquated street lighting to energy-efficient LED bulbs. My role was created to promote and sell the energy-efficient bulbs, while touting the long-term advantage of reduced energy costs. I had to develop a way to educate city light officials on the value of our energy-efficient bulbs—which was a challenge since our products had an expensive up-front cost compared to less efficient lighting options. I created an information packet and held local community events aimed at city officials and the tax-paying public. There, I was able to demo the company*

*product, answer questions, and evangelize the value of LED bulbs for the long term. It was crucial to have the public on board and I was able to reach a wide variety of community members with these events. I not only reached my first-year sales goal of \$100,000, but I was also able to help us land another contract in a neighboring city. Plus, the community-focused strategy garnered attention from the national media. And I'm proud to say I got a promotion within one year to senior sales representative."*

**Read More:** [The Perfect Formula for Answering “What Is Your Greatest Accomplishment” in an Interview](#)

## 11. Tell me about a challenge or conflict you've faced at work, and how you dealt with it.

You're probably not eager to talk about conflicts you've had at work during a job interview. But if you're asked directly, don't pretend you've never had one. Be honest about a difficult situation you've faced (but without going into the kind of detail you'd share venting to a friend). "Most people who ask are only looking for evidence that you're willing to face these kinds of issues head-on and make a sincere attempt at coming to a resolution," former recruiter Richard Moy says. Stay calm and professional as you tell the story (and answer any follow-up questions), spend more time talking about the resolution than the conflict, and mention what you'd do differently next time to show "you're open to learning from tough experiences."

### Possible answer to "Tell me about a challenge or conflict you've faced at work, and how you dealt with it."

*"Funnily enough, last year I was part of a committee that put together a training on conflict intervention in the workplace and the amount of pushback we got for requiring attendance really put our training to the test. There was one senior staff member in particular who seemed adamant. It took some careful listening to understand he felt like it wasn't the best use of his time given the workload he was juggling. I made sure to acknowledge his concern. And then I focused on his direct objection and explained how the training was meant to improve not just the culture of the company, but also the efficiency at which we operated—and that the goal was for the training to make everyone's workload feel lighter. He did eventually attend and was there when I talked to the whole staff about identifying the root issue of a conflict and addressing that directly"*

*without bringing in other issues, which is how I aim to handle any disagreement in the workplace.”*

**Read More:** [3 Ways You’re Messing Up the Answer to “Tell Me About a Conflict You’ve Faced at Work”](#)

## 12. Tell me about a time you demonstrated leadership skills.

You don't have to have a fancy title to act like a leader or demonstrate leadership skills. Think about a time when you headed up a project, took the initiative to propose an alternate process, or helped motivate your team to get something done. Then use the STAR method to tell your interviewer a story, giving enough detail to paint a picture (but not so much that you start rambling) and making sure you spell out the result. In other words, be clear about why you're telling this particular story and connect all the dots for the interviewer.

### Possible answer to “Tell me about a time you demonstrated leadership skills.”

*“I think that a good leader is someone who can make decisions while also listening to others and being willing to admit when you’re wrong and course correct. In my last role, my team and I were responsible for giving a big presentation to a prospective client. I quickly assigned different tasks to members of my team, but the project never really got moving. I gave everyone an opportunity to share their input and concerns, and it turned out that they were struggling in the roles I’d given them. I ended up switching a few people around. Meanwhile, the employee I’d assigned to give the presentation was nervous, but still wanted to give it a try. I worked with them to make sure they were ready and even held a practice session so that they could rehearse in a more comfortable environment. When the time came for the real thing, they nailed it! We landed the client and the company still has the account to this day. And that employee became a go-to person for important client presentations. I’m really glad I took the time to listen to everyone’s concerns so that I could re-evaluate my approach and help my team be the best it could be.”*

**Read More:** [The Best Way to Answer “Tell Me About a Time You Demonstrated Leadership Skills” in a Job Interview](#)

## 13. What's a time you disagreed with a decision that was made at work?

The ideal anecdote here is one where you handled a disagreement professionally and learned something from the experience. [Zhang recommends](#) paying particular attention to how you start and end your response. To open, make a short statement to frame the rest of your answer, one that nods at the ultimate takeaway or the reason you're telling this story. For example: "I learned early on in my professional career that it's fine to disagree if you can back up your hunches with data." And to close strong, you can either give a one-sentence summary of your answer ("In short...") or talk briefly about how what you learned or gained from this experience would help you in the role you're interviewing for.

### Possible answer to "What's a time you disagreed with a decision that was made at work?"

*"In my job as a finance assistant, I was in charge of putting together reports for potential company investments. It was important to get the details and numbers right so that leaders had the best information to make a decision. One time, my boss asked me to generate a new report on a Wednesday morning and wanted it done by Thursday at 5 PM. Because I'm committed to high-quality work and I wasn't sure my boss fully understood what goes into each report, I knew I needed to speak up. At her next available opening, I sat down with my boss and explained my concerns. She was firm that the report would be completed by Thursday at 5 PM. So I decided to ask if there was anyone who could help out. After thinking about it, my boss found another assistant who could put in a few hours. While it was a tight timeline, we got the report done, and the committee was really pleased to review it at the meeting. My boss appreciated my extra efforts to make it happen and I felt good that I hadn't let the quality of the report slip. It was a good experience of being a team player but also knowing when and how to ask for help. And once I explained how much time and work goes into each report, my boss was careful to assign them further in advance."*

**Read More:** [Here's the Secret to Answering "Tell Me About a Time You Had a Conflict With Your Boss" in an Interview](#)

## 14. Tell me about a time you made a mistake.

You're probably not too eager to dig into past blunders when you're trying to impress an interviewer and land a job. But talking about a mistake and winning someone over aren't mutually exclusive, Moy says. In fact, if you do it right, it can *help* you. The key is to be honest without placing blame on other people, then explain what you learned from your mistake and what actions you took to ensure it didn't happen again. At the end of the day, employers are looking for folks who are self-aware, can take feedback, and care about doing better.

## Possible answer to "Tell me about a time you made a mistake."

*"Early in my career, I missed a deadline that ended up costing us a really big account. There were a lot of factors that contributed to this, but ultimately, I was the one who dropped the ball. From that experience, I went back and thought really hard about what I could've controlled and what I would've changed. It turns out that I was not nearly as organized as I thought I was. I sat down with my boss, asked for suggestions on how to improve my organizational skills, and a few months later I was able to score an even bigger account for the department."*

**Read More:** [3 Rules That Guarantee You'll Nail the Answer to "Tell Me About a Time You Made a Mistake"](#)

## 15. Tell me about a time you failed.

This question is very similar to the one about making a mistake, and you should approach your answer in much the same way. Make sure you pick a real, actual failure you can speak honestly about. Start by making it clear to the interviewer how you define failure. For instance: "As a manager, I consider it a failure whenever I'm caught by surprise. I strive to know what's going on with my team and their work." Then situate your story in relation to that definition and explain what happened. Finally, don't forget to share what you learned. It's OK to fail—everyone does sometimes—but it's important to show that you took something from the experience.

## Possible answer to "Tell me about a time you failed."

*"As a team manager, I consider it a failure if I don't know what's going on with my staff and their work—basically if a problem catches me by surprise then I've failed somewhere along the way. Even if the outcome is ultimately fine, it means I've left a team member unsupported at some point. A somewhat recent example would be this training we do*

*every year for new project managers. Because it's an event that my team has run so many times, I didn't think to check in and had no idea a scheduling conflict was brewing into a full-on turf war with another team. The resolution actually ended up being a quick and easy conversation at the leadership team meeting, but had I just asked about it sooner it would never have been a problem to begin with. I definitely learned my lesson about setting reminders to check in about major projects or events even if they've been done dozens of times before."*

**Read More:** [4 Steps for Answering “Tell Me About a Time When You Failed”](#)

## 16. Why are you leaving your current job?

This is a toughie, but one you can be sure you'll be asked. Definitely keep things positive—you have nothing to gain by being negative about your current employer. Instead, frame things in a way that shows that you're eager to take on new opportunities and that the role you're interviewing for is a better fit for you. For example, "I'd really love to be part of product development from beginning to end, and I know I'd have that opportunity here." And if you were let go from your most recent job? Keep it simple: "Unfortunately, I was let go," is a totally acceptable answer.

### Possible answer to "Why are you leaving your current job?"

*"I'm ready for the next challenge in my career. I loved the people I worked with and the projects I worked on, but at some point I realized I wasn't being challenged the way I used to be. Rather than let myself get too comfortable, I decided to pursue a position where I can continue to grow."*

**Read More:** [4 Better Ways to Answer “Why Are You Leaving Your Job?”](#)

## 17. Why were you fired?

Of course, they may ask the follow-up question: *Why* were you let go? If you lost your job due to layoffs, you can simply say, "The company [reorganized/merged/was acquired] and unfortunately my [position/department] was eliminated." But what if you were fired for performance reasons? Your best bet is to be honest (the job-seeking world is small, after all). But it doesn't have to be a deal breaker. Frame it as a learning experience: Share how

you've grown and how you approach your job and life now as a result. And if you can portray your growth as an advantage for this next job, even better.

## Possible answer to "Why were you fired?"

*"After working for XYZ Inc. for four years, there were some changes made to the amount of client calls we were expected to process per hour. I used the techniques we were taught after the change took effect, but didn't want our customer service to slip. Unfortunately, I wasn't consistently completing the required number of calls, and, as a result, I was let go. I felt really bad about this and in retrospect I could have done better sticking to the process that would have let me meet the per hour quota. But you've told me about the customer service standards and the volume expectations here, and I believe it won't be a problem."*

**Read More:** [Stop Cringing! How to Tell an Interviewer You've Been Fired](#)

## 18. Why was there a gap in your employment?

Maybe you were taking care of children or aging parents, dealing with health issues, or traveling the world. Maybe it just took you a long time to land the right job. Whatever the reason, you should be prepared to discuss the gap (or gaps) on your resume. Seriously, practice saying your answer out loud. The key is to be honest, though that doesn't mean you have to share more details than you're comfortable with. If there are skills or qualities you honed or gained in your time away from the workforce—whether through volunteer work, running a home, or responding to a personal crisis—you can also talk about how those would help you excel in this role.

## Possible answer to "Why was there a gap in your employment?"

*"I spent a number of years working at a company in a very demanding job, in which—as you'll see from my references—I was very successful. But I'd reached a stage in my career where I wanted to focus on my personal growth. The time I spent traveling taught me a lot about how to get along with people of all ages and cultures. Now I feel more than ready to jump back into my career with renewed energy and focus and I feel this role is the ideal way to do that."*

**Read More:** [How to Explain the Gap in Your Resume With Ease](#)

## 19. Can you explain why you changed career paths?

Don't be thrown off by this question—just take a deep breath and explain to the hiring manager why you've made the career decisions you have. More importantly, give a few examples of how your past experience is [transferable](#) to the new role. This doesn't have to be a direct connection; in fact, it's often more impressive when a candidate can show how seemingly irrelevant experience is very relevant to the role.

### Possible answer to "Can you explain why you changed career paths?"

*"Ever since my brother was diagnosed with a heart condition, I've been training and running with him in your annual Heart Run to raise money for your organization and help support patients with expenses not covered by insurance. Each time, I've been struck by how truly dedicated and happy to be there your employees have been. So when I saw this posting for a fundraising role, it felt like it was meant to be. For the last 10 years of my career I've been an account executive for various SaaS companies, and I've really honed my skills when it comes to convincing organizations to make regular payments for something over the long-term. But I've been looking for a position in fundraising where I can use these skills to really help people and I'm highly motivated to do that with your organization."*

**Read More:** [How to Explain Your Winding Career Path to a Hiring Manager](#)

## 20. What's your current salary?

It's now illegal for some or all employers to ask you about your salary history in several cities and states, including New York City; Louisville, North Carolina; California; and Massachusetts. But no matter where you live, it can be stressful to hear this question. Don't panic—there are several possible strategies you can turn to. For example, you can deflect the question, Muse career coach Emily Liou says, with a response like: "Before discussing any salary, I'd really like to learn more about what this role entails. I've done a lot of research on [Company] and I am certain if it's the right fit, we'll be able to agree on a number that's fair and competitive to both parties." You can also reframe the question around your salary expectations or requirements (see question 38) or choose to share the number if you think it will work in your favor.

### Possible answer to "What's your current salary?"

*"Before discussing any salary, I'd really like to learn more about what this role entails. I've done a lot of research on [Company] and I am certain if it's the right fit, we'll be able to agree on a number that's fair and competitive to both parties."*

**Read More:** [Here's How You Answer the Illegal "What's Your Current Salary" Question](#)

## 21. What do you like least about your job?

Tread carefully here! The last thing you want to do is let your answer devolve into a rant about how terrible your current company is or how much you hate your boss or that one coworker. The easiest way to handle this question with poise is to focus on an opportunity the role you're interviewing for offers that your current job doesn't. You can keep the conversation positive and emphasize why you're so excited about the job.

### Possible answer to "What do you like least about your job?"

*"In my current role, I'm responsible for drafting media lists to pitch. While I've developed a knack for this and can do it when it is necessary, I'm looking forward to a job that allows me to have a more hands-on role in working with media partners. That's one of the things that most excited me about your account supervisor position."*

**Read More:** [What Interviewers Really Want When They Ask, "What Do You Like Least About Your Job?"](#)

## 22. What are you looking for in a new position?

Hint: Ideally the same things that this position has to offer. Be specific.

### Possible answer to "What are you looking for in a new position?"

*"I've been honing my data analysis skills for a few years now and, first and foremost, I'm looking for a position where I can continue to exercise those skills. Another thing that's important to me is the chance to present my findings and suggestions directly to clients. I'm always very motivated by being able to see the impact of my work on other people. And I'm definitely looking for a position where I can grow since I hope to take on managerial responsibilities in the future. To sum it up, I'd love a position where I can use my skills to make an impact that I can see with my own eyes. Of course, the position is only part of the equation. Being at a company where I can grow and work toward*

*something I care about matters, too. DNF's goal of being at the intersection between data and education inspires me, and I'm really excited about this opportunity."*

**Read More:** [4 Steps for Answering "What Are You Looking for in a New Position?"](#)

## 23. What type of work environment do you prefer?

Hint: Ideally one that's similar to the environment of the company you're applying to. Be specific.

### Possible answer to "What type of work environment do you prefer?"

*"I really like the environment in my current position. My manager is a great resource and always willing to help out when I run into an issue, but they trust me to get my work done so I have a lot of freedom in how I schedule and prioritize, which is very important to me. Everyone has their own cubicle, so it's often pretty quiet to get our work done, but we all get lunch together and our team has a lot of check-in meetings and communicates frequently via Slack so we still get a lot of opportunities to bounce ideas off each other. So I like both individual and more collaborative work. How would you describe the mix here?"*

**Read More:** [3 Steps to Answering "What Type of Work Environment Do You Prefer?"](#)

## 24. What's your work style?

When an interviewer asks you about your work style, they're probably trying to imagine you in the role. How will you approach your work? What will it be like to work with you? Will you mesh well with the existing team? You can help them along by choosing to focus on something that's important to you *and* aligns with everything you've learned about the role, team, and company so far. The question is broad, which means you have a lot of flexibility in how you answer: You might talk about how you communicate and collaborate on cross-functional projects, what kind of remote work setup allows you to be most productive, or how you approach leading a team and managing direct reports. Just try to keep it positive. And remember, telling a story will almost always make your answer more memorable.

### Possible answer to "What's your work style?"

*"I tend to do my best work when I'm collaborating with colleagues and we're working together toward a common goal. I was that rare student who loved group projects and now I still get a rush of excitement when I'm planning marketing campaigns with a team and bringing new and different voices into the fold. When I was working at XYZ Agency, I made it a habit to extend invitations to folks in different departments to join certain brainstorming and feedback sessions. Some of our most successful campaigns grew out of the ideas we generated together with coworkers in IT, HR, product, and customer success. That's why I was so excited to learn that this role would have me working closely with the product and sales teams as well as with a talented marketing team. The other thing I find is crucial to making these collaborations successful is organization and documentation, so I'm also really big on creating one central home for all materials related to a project, including meeting notes, action items, drafts of campaign copy and visuals, and timelines."*

**Read More:** [How to Answer “What Is Your Work Style?” in an Interview \(Plus Examples!\)](#)

## 25. What's your management style?

The best managers are strong but flexible, and that's exactly what you want to show off in your answer. (Think something like, "While every situation and every team member requires a bit of a different strategy, I tend to approach my employee relationships as a coach...") Then share a couple of your best managerial moments, like when you grew your team from five to 15 or coached an underperforming employee to become the company's top salesperson.

### Possible answer to "What's your management style?"

*"Management style is so hard to put your finger on, but I think in general a good manager gives clear directions and actually stays pretty hands-off, but is ready and available to jump in to offer guidance, expertise, and help when needed. I try my best to make that my management style. I also go out of my way to make sure I know when my team needs help. That means plenty of informal check-ins, both on the work they're doing and on their general job satisfaction and mental well-being. I remember one project in particular at my most recent position that involved everyone working on a separate aspect of the product. This meant a lot of independent work for my team of seven people, but rather than bog everyone down with repetitive meetings to update me and everyone*

*else on progress made, I created a project wiki that allowed us to communicate new information when necessary without disrupting another team member's work. I then made it my job to make sure no one was ever stuck on a problem too long without a sounding board. Ultimately, despite the disparate project responsibilities, we ended up with a very cohesive product and, more importantly, a team that wasn't burnt out."*

**Read More:** [How to Answer "What's Your Management Style?"](#)

## 26. How would your boss and coworkers describe you?

First, be honest (remember, if you make it to the final round, the hiring manager will be calling your former bosses and coworkers for references!). Then try to pull out strengths and traits you haven't discussed in other aspects of the interview, such as your strong work ethic or your willingness to pitch in on other projects when needed.

### Possible answer to "How would your boss and coworkers describe you?"

*"Actually, in my most recent performance review in April, my direct supervisor described me as someone who takes initiative and doesn't shy away from hard problems. My role involves a lot of on-site implementation, and when things go wrong, it's usually up to me to fix it. Rather than punting the problem back to the team, I always try to do what I can first. I know she appreciates that about me."*

**Read More:** [3 Strategies for Answering "How Would Your Boss or Coworkers Describe You?"](#)

## 27. How do you deal with pressure or stressful situations?

Here's another question you may feel the urge to sidestep in an effort to prove you're the perfect candidate who can handle anything. But it's important not to dismiss this one (i.e. don't say, "I just put my head down and push through it," or, "I don't get stressed out"). Instead, talk about your go-to strategies for dealing with stress (whether it's meditating for 10 minutes every day or making sure you go for a run or keeping a super-detailed to-do list) and how you communicate and otherwise proactively try to mitigate pressure. If you can give a real example of a stressful situation you navigated successfully, all the better.

## Possible answer to “How do you deal with pressure or stressful situations?”

*“I stay motivated by thinking about the end result. I’ve found that even in the midst of a challenging situation, reminding myself of my goals helps me take a step back and stay positive.”*

**Read More:** [3 Ways You’re Messing Up the Answer to “How Do You Deal With Stressful Situations?”](#)

## 28. What do you like to do outside of work?

Interviewers will sometimes ask about your hobbies or interests outside of work in order to get to know you a little better—to find out what you’re passionate about and devote time to during your off-hours. It’s another chance to let your personality shine. Be honest, but keep it professional and be mindful of answers that might make it sound like you’re going to spend *all* your time focusing on something other than the job you’re applying for.

### Possible answer to “What do you like to do outside of work?”

*“I’m a huge foodie. My friends and I love trying new restaurants in town as soon as they open—the more unusual the better! I love discovering new foods and cuisines, and it’s also a great activity to share with friends. I try to go out with the same group at least once a week and it’s a fun way to make sure we keep in touch and share experiences even when we’re busy with other things. We even took a trip to New York City and spent each day in a different neighborhood, buying something to share from a few restaurants.”*

**Read More:** [How to Answer “What Are Your Hobbies?” in an Interview \(It’s Not a Trick Question!\)](#)

## 29. Are you planning on having children?

Questions about your family status, gender (“How would you handle managing a team of all men?”), nationality (“Where were you born?”), religion, or age are illegal—but they still get asked (and frequently). Of course, not always with ill intent—the interviewer might just be trying to make conversation and might not realize these are off-limits—but you should

definitely tie any questions about your personal life (or anything else you think might be inappropriate) back to the job at hand.

## Possible answer to “Are you planning on having children?”

*“You know, I’m not quite there yet. But I am very interested in the career paths at your company. Can you tell me more about that?”*

**Read More:** [5 Illegal Interview Questions and How to Dodge Them](#)

## 30. How do you stay organized?

Would you want to work with a hot mess? Yeah, we didn’t think so. Neither does anyone else. A disorganized worker doesn’t just struggle in their own role, they can also create chaos for peers, managers, direct reports, clients, customers, and anyone else they interact with. So interviewers will often ask about how you keep yourself organized to make sure you’d be able to handle the workload and gauge what you’d be like to work with. In your answer, you’ll want to reassure them you’d have things under control (both in what you say and how you say it), describe a specific system or method you’ve used (bonus points if you can tie it to the role you’re interviewing for), and explain how it benefited you and your team. Just make sure your answer is succinct and, well, organized.

## Possible answer to “How do you stay organized?”

*“I take pride in my ability to stay organized, and it’s really come in handy in my past roles and especially the social media assistant job I’m in now. First, I keep a really meticulous calendar for each of the platforms I’m responsible for using Hootsuite—which I noticed you use here as well—and I try to block off time twice a week to get ahead on creating and slotting in posts.*

*“Second, I’m a big fan of Trello, where I have one personal board I use as a to-do list color-coded by type of task and marked with priority level and one shared marketing team board that we use to coordinate campaigns launching across social, email, and other channels. We pay very close attention to the news in case we need to pause a campaign. If needed, I’d tag all the relevant stakeholders on Trello, immediately suspend all scheduled content in Hootsuite, and start a discussion on Slack or suggest a meeting to reassess strategy.*

*“Finally, I created a shared folder on Google Drive with subfolders by campaign that I update with one-pagers on goals and strategies, assets, a record of the actual posts deployed, performance analyses, and retros. That way, there’s a go-to place for anyone on the team to refer back to past projects, which I’ve found really helps us learn from every campaign and incorporate those learnings into what we’re working on next.”*

**Read More:** [What Interviewers Really Want to Know When They Ask “How Do You Stay Organized?”](#)

## 31. How do you prioritize your work?

Your interviewers want to know that you can manage your time, exercise judgement, communicate, and shift gears when needed. Start by talking about whatever system you've found works for you to plan your day or week, whether it's a to-do list app you swear by or a color-coded spreadsheet. This is one where you'll definitely want to lean on a real-life example. So go on to describe how you've reacted to a last-minute request or another unexpected shift in priorities in the past, incorporating how you evaluated and decided what to do and how you communicated with your manager and/or teammates about it.

### Possible answer to “How do you prioritize your work?”

*“I’d be lost without my daily to-do list! At the beginning of each workday, I write out tasks to complete, and list them from highest to lowest priority to help keep me on track. But I also realize priorities change unexpectedly. On one particular day recently, I had planned to spend most of my time making phone calls to advertising agencies to get price quotes for an upcoming campaign. Then I did a quick check-in with my manager. She mentioned she needed help putting together a presentation ASAP for a major potential client. I moved the more flexible task to the end of the week and spent the next few hours updating the time-sensitive presentation. I make it a point to keep lines of communication open with my manager and coworkers. If I’m working on a task that will take a while to complete, I try to give a heads-up to my team as soon as possible. If my workload gets to be unmanageable, I check in with my boss about which items can drop to the bottom of the priority list, and then I try to reset expectations about different deadlines.”*

**Read More:** [A Foolproof Method to Answer the Interview Question “How Do You Prioritize Your Work?”](#)

## 32. What are you passionate about?

You're not a robot programmed to do your work and then power down. You're a human, and if someone asks you this question in an interview, it's probably because they want to get to know you better. The answer can align directly with the type of work you'd be doing in that role—like if, for example, you're applying to be a graphic designer and spend all of your free time creating illustrations and data visualizations to post on Instagram.

But don't be afraid to talk about a hobby that's different from your day-to-day work. Bonus points if you can "take it one step further and connect how your passion would make you an excellent candidate for the role you are applying for," says Muse career coach Al Dea. Like if you're a software developer who loves to bake, you might talk about how the ability to be both creative and precise informs your approach to code.

### Possible answer to "What are you passionate about?"

*"One of my favorite pastimes is knitting—I love being able to create something beautiful from nothing. Of course, knitting also requires a keen attention to detail and a lot of patience. Luckily, as an accountant I have cultivated both of those qualities!"*

**Read More:** [3 Authentic Ways to Answer "What Are You Passionate About?" in a Job Interview](#)

## 33. What motivates you?

Before you panic about answering what feels like a probing existential question, consider that the interviewer wants to make sure you're excited about *this* role at *this* company, and that you'll be motivated to succeed if they pick you. So think back to what has energized you in previous roles and pinpoint what made your eyes light up when you read this job description. Pick one thing, make sure it's relevant to the role and company you're interviewing for, and try to weave in a story to help illustrate your point. If you're honest, which you should be, your enthusiasm will be palpable.

### Possible answer to "What motivates you?"

*"I'm driven primarily by my desire to learn new things—big or small—and take on new responsibilities so that I'm constantly growing as an employee and contributing more to my team and organization. I spent several summers working as a camp counselor and*

*felt most fulfilled when I volunteered to lead planning for a talent show, jumped in to help with scheduling logistics, and learned how to run pickups efficiently. All of that experience helped immensely when I took a step up to become the lead counselor last year focused on operations, and that's what excites me so much about the opportunity to take on this managerial role for the after-school program.”*

**Read More:** [5 Easy Steps to Answer “What Motivates You?” in an Interview](#)

## 34. What are your pet peeves?

Here's another one that feels like a minefield. But it'll be easier to navigate if you know why an interviewer is asking it. Most likely, they want to make sure you'll thrive at their company—and get a glimpse of how you deal with conflict. So be certain you pick something that doesn't contradict the culture and environment at this organization while still being honest. Then explain why and what you've done to address it in the past, doing your best to stay calm and composed. Since there's no need to dwell on something that annoys you, you can keep this response short and sweet.

### Possible answer to “What are your pet peeves?”

*“It bothers me when an office’s schedule is really disorganized, because in my experience, disorganization can cause confusion, which can hurt the motivation of the team. As a person who likes things to be orderly, I try to help keep my team on task while also allowing for flexibility.”*

**Read More:** [6 Tips for Answering “What Are Your Pet Peeves?” in an Interview](#)

## 35. How do you like to be managed?

This is another one of those questions that's about finding the right fit—both from the company's perspective and your own. Think back on what worked well for you in the past and what didn't. What did previous bosses do that motivated you and helped you succeed and grow? Pick one or two things to focus on and always articulate them with a positive framing (even if your preference comes from an experience where your manager behaved in the opposite way, phrase it as what you *would* want a manager to do). If you can give a positive example from a great boss, it'll make your answer even stronger.

## Possible answer to “How do you like to be managed?”

*“I enjoy having my hands in a lot of different projects, so I like working with managers who allow their employees to experiment, be independent, and work cross-functionally with other teams. At the same time, I really welcome it when a boss provides me with support, guidance, and coaching. No one can do anything alone, and I believe when managers and employees collaborate together and learn from one another everyone comes out on top.”*

**Read More:** [3 Easy Steps to Answer “How Do You Like to Be Managed?” in an Interview](#)

## 36. Do you consider yourself successful?

This question might make you uncomfortable. But you can think of it as an opportunity to allow the interviewer to get to know you better and to position yourself as an excellent choice for this job. First off, make sure you say yes! Then pick one specific professional achievement you’re proud of that can be tied back to the role you’re interviewing for—one that demonstrates a quality, skill, or experience that would help you excel in this position. You’ll want to explain *why* you consider it a success, talk about the process in addition to the outcome, and highlight your own accomplishment without forgetting your team.

Zooming in on one story will help if you feel awkward tooting your own horn!

## Possible answer to “Do you consider yourself successful?”

*“I do consider myself successful, even though I’m early in my professional career. I took a full load of classes in my junior year of college because I wanted to take that summer to volunteer for a human rights organization overseas. I knew that I needed to make sure I was on track with my major, minor, and graduation requirements. It was difficult to juggle it all with my part-time job, which I kept to help account for the fact that I wouldn’t be earning money over the summer, and there were a few sleepless nights. But it was worth the hard work: I ended the year with a 3.9 GPA and the opportunity to volunteer for the agency in Ghana without falling behind my graduation timeline. For me success is about setting a goal and sticking with it, no matter how hard it is, and this experience was proof that I could be successful even when there’s a lot to balance, which I know there always is at a nonprofit like this one.”*

**Read More:** [How to Answer “Do You Consider Yourself Successful?” Without Feeling Like a Show-Off](#)

## 37. Where do you see yourself in five years?

If asked this question, be honest and specific about your future goals, but consider this: A hiring manager wants to know a) if you've set realistic expectations for your career, b) if you have ambition (a.k.a., this interview isn't the first time you're considering the question), and c) if the position aligns with your goals and growth. Your best bet is to think realistically about where this position could take you and answer along those lines. And if the position isn't necessarily a one-way ticket to your aspirations? It's OK to say that you're not quite sure what the future holds, but that you see this experience playing an important role in helping you make that decision.

### Possible answer to "Where do you see yourself in five years?"

*"In five years, I'd like to be in a position where I know more about my longer-term career aspirations as a designer. I will have gotten experience working for a design agency and know more about the industry overall. I'll have grown my technical skills and learned how to take feedback from clients and incorporate it. And the way your agency is set up, I'll also have gotten the opportunity to design different kinds of deliverables—including websites, branding, and ad campaigns—for different kinds of clients to see where I really feel at home before settling on a focus."*

**Read More:** [How to Answer "Where Do You See Yourself in 5 Years?"](#)

## 38. How do you plan to achieve your career goals?

Having goals shows interviewers you care, are ambitious, and can think ahead. Having a plan for how you'll achieve your goals demonstrates your self-motivation as well as organizational and time management skills. Finally, the fact that you've accomplished past goals you've set for yourself is proof of your ability to follow through. All together, these are indications that you can not only set and achieve goals of your own, but also help your prospective boss, team, and company do the same. To craft your answer, make sure you focus on one or two goals in detail, explain why the goals are meaningful, communicate what milestones are coming up, highlight past successes, and connect back to *this* job.

### Possible answer to "How do you plan to achieve your career goals?"

*"My current goal is to earn the CPA license so that I'm fully certified and prepared to contribute in a junior staff accounting job. My undergraduate degree is in finance and I*

*completed an accounting internship with XYZ Company last summer. While I was there, I decided that each week I'd ask one person from a different team to coffee to learn about their job and career path. Not only did those conversations impress upon me the importance of getting my CPA as soon as possible, they also helped me realize I was eager to pursue forensic accounting, which is why I'm so excited about the opportunity to join this team. In order to ensure I earn my CPA this year, I enrolled in NASBA workshops, created a study schedule to keep myself on track, and will be taking my first trial test in three weeks. I plan on taking the actual test within the next three to six months."*

**Read More:** [How to Answer “How Do You Plan to Achieve Your Career Goals?” in an Interview](#)

## 39. What are your career aspirations?

Career aspirations are bigger and loftier than career goals. With this question, interviewers are asking: What kind of career would make you happiest (while also being realistic)? Your aspirations might revolve around what kind of company you'd like to work for, what tasks you'd like to do, who you'd like to help, or how you'd like to be seen by your colleagues. So to answer this question, talk about what would energize and fulfill you and connect it to the position you're interviewing for. Be specific about how this job will help you achieve your career aspirations.

### Possible answer to “What are your career aspirations?”

*“After growing up in a food desert, my biggest professional aspiration is to help make healthy food more widely available and accessible regardless of where you live. I also love solving complex problems. Currently, as a project manager, I specialize in strategic planning and combine it with a natural ability to engage critical stakeholders—resulting in on-time and under-budget delivery. This role would help me use those skills to work on a mission I’m passionate about. I am determined to use these skills to help your organization guarantee our community has access to affordable, nutritious food and information to make healthy decisions. In the next five or so years, I would love to take on additional responsibility and be in a decision-making role to drive the mission beyond our community and support even more families in gaining access to nutritious food options.”*

**Read More:** [How to Answer “What Are Your Career Aspirations?” in an Interview](#)

## 40. What's your dream job?

Along similar lines, the interviewer wants to uncover whether this position is really in line with your ultimate career goals. While “an NBA star” might get you a few laughs, a better bet is to talk about your goals and ambitions—and why this job will get you closer to them.

**Read More:** [The Secret Formula to Answering “What's Your Dream Job?” in an Interview](#)

## 41. What other companies are you interviewing with?

Companies might ask you who else you’re interviewing with for a few reasons. Maybe they want to see how serious you are about this role and team (or even this field) or they’re trying to find out who they’re competing with to hire you. On one hand, you want to express your enthusiasm for this job, but at the same time, you don’t want to give the company any more leverage than it already has by telling them there’s no one else in the running. Depending on where you are in your search, you can talk about applying to or interviewing for a few roles that have XYZ in common—then mention how and why *this* role seems like a particularly good fit.

### Possible answer to “What other companies are you interviewing with?”

*“I’m interviewing with a few companies for a range of positions, but they all come down to delivering an excellent customer experience. I wanted to keep an open mind about how to best achieve that goal, but so far it seems that this role will really allow me to focus all of my energy on customer experience and retention, which I find very appealing.”*

**Read More:** [How to Answer “What Other Companies Are You Interviewing With?”](#)

## 42. What makes you unique?

“They genuinely want to know the answer,” Dea promises. Give them a reason to pick you over other similar candidates. The key is to keep your answer relevant to the role you’re applying to. So the fact that you can run a six-minute mile or crush a trivia challenge might not help you get the job (but hey, it depends on the job!). Use this opportunity to tell them something that would give you an edge over your competition for *this* position. To figure out what that is, you can ask some former colleagues, think back to patterns you’ve seen in

feedback you get, or try to distill why people tend to turn to you. Focus on one or two things and don't forget to back up whatever you say with evidence.

## Possible answer to “What makes you unique?”

*“I basically taught myself animation from scratch. I was immediately drawn to it in college, and with the limited resources available to me, I decided to take matters into my own hands—and that’s the approach I take in all aspects of my work as a video editor. I don’t just wait around for things to happen, and when I can, I’m always eager to step in and take on new projects, pick up new skills, or brainstorm new ideas.”*

**Read More:** [A Simple Way to Answer “What Makes You Unique?” in Your Job Search \(Plus, Examples!\)](#)

## 43. What should I know that’s not on your resume?

It's a good sign if a recruiter or hiring manager is interested in *more* than just what's on your resume. It probably means they looked at your resume, think you might be a good fit for the role, and want to know more about you. To make this wide-open question a little more manageable, try talking about a positive trait, a story or detail that reveals a little more about you and your experience, or a mission or goal that makes you excited about this role or company.

## Possible answer to “What should I know that’s not on your resume?”

*“Well, one thing you won’t find on my resume: the time I had to administer emergency CPR. Last year, I was at the lake when I saw a young girl who looked like she was drowning. I was a lifeguard in high school, so I swam out, brought her to shore, and gave her CPR. Although this was—hopefully—a one-time event, I’ve always been able to stay calm during stressful situations, figure out a solution, and then act. As your account manager, I’d use this trait to quickly and effectively resolve issues both within the team and externally. After all, obstacles are inevitable, especially in a startup environment. And if anyone needs CPR at the office beach party, well, I’m your woman.”*

**Read More:** [The Right Way to Answer “What Should I Know That’s Not on Your Resume?”](#)

## 44. What would your first few months look like in this role?

Your potential future boss (or whoever else has asked you this question) wants to know that you've done your research, given some thought to how you'd get started, and would be able to take initiative if hired. (In some interviews, you might even get the more specific, "What would your first 30, 60, or 90 days look like in this role?") So think about what information and aspects of the company and team you'd need to familiarize yourself with and which colleagues you'd want to sit down and talk to. You can also suggest one possible starter project to show you'd be ready to hit the ground running and contribute early on. This won't necessarily be the thing you do first if you do get the job, but a good answer shows that you're thoughtful and that you care.

### Possible answer to "What would your first few months look like in this role?"

*"It's been exciting to hear about some of the new initiatives the company has started in our previous conversations—like the database project and the company-wide sync, but I know there's still a lot for me to learn. The first thing I'd do is line up meetings with the stakeholders involved in the projects I'd be tackling to help me figure out what I don't know and then go from there. Hopping into a database project halfway through can be tricky, but I'm confident that once I know what all the stakeholders are looking for, I'll be able to efficiently plot out our next steps and set appropriate deadlines. From there, I'll be focused on hitting the milestones that I've set for the team."*

**Read More:** [The 30-60-90 Day Plan: Your Secret Weapon for New Job Success](#)

## 45. What are your salary expectations?

The number one rule of answering this question is: Figure out your salary requirements ahead of time. Do your research on what similar roles pay by using sites like PayScale and reaching out to your network. Be sure to take your experience, education, skills, and personal needs into account, too! From there, Muse career coach [Jennifer Fink](#) suggests choosing from one of three strategies:

- Give a salary range: But keep the bottom of your stated range toward the mid-to-high point of what you're actually hoping for, Fink says.

- Flip the question: Try something like “That’s a great question—it would be helpful if you could share what the range is for this role,” Fink says.
- Delay answering: Tell your interviewer that you’d like to learn more about the role or the rest of the compensation package before discussing pay.

(And here’s some more info on responding to a question about your [salary requirements on an application form](#).)

## Possible answer to “What are your salary expectations?”

*“Taking into account my experience and Excel certifications, which you mentioned earlier would be very helpful to the team, I’m looking for somewhere between \$42,000 and \$46,000 annually for this role. But for me, benefits definitely matter as well. Your free on-site gym, the commuter benefits, and other perks could definitely allow me to be a bit flexible with salary.”*

**Read More:** [3 Strategies for Answering “What Are Your Salary Expectations?” in an Interview](#)

## 46. What do you think we could do better or differently?

This question can really do a number on you. How do you give a meaty answer without insulting the company or, worse, the person you’re speaking with? Well first, take a deep breath. Then start your response with something positive about the company or specific product you’ve been asked to discuss. When you’re ready to give your constructive feedback, give some background on the perspective you’re bringing to the table and explain why you’d make the change you’re suggesting (ideally based on some past experience or other evidence). And if you end with a question, you can show them you’re curious about the company or product and open to other points of view. Try: “Did you consider that approach here? I’d love to know more about your process.”

**Read More:** [How to Answer the “How Would You Improve Our Company?” Interview Question Without Bashing Anyone](#)

## 47. When can you start?

Your goal here should be to set realistic expectations that will work for both you and the company. What exactly that sounds like will depend on your specific situation. If you're ready to start immediately—if you're unemployed, for example—you could offer to start within the week. But if you need to give notice to your current employer, don't be afraid to say so; people will understand and respect that you plan to wrap things up right. It's also legitimate to want to take a break between jobs, though you might want to say you have "previously scheduled commitments to attend to" and try to be flexible if they really need someone to start a bit sooner.

## Possible answer to "When can you start?"

*"I am excited for the opportunity to join your team. I have several projects to wrap up in my current role at [Company]. I plan to give them two weeks' notice to make a smooth transition for my coworkers and will be happy to come onboard with the team here after that time."*

**Read More:** [4 Ways to Answer the Interview Question "When Can You Start?"](#)

## 48. Are you willing to relocate?

While this may sound like a simple yes-or-no question, it's often a little bit more complicated than that. The simplest scenario is one where you're totally open to moving and would be willing to do so for this opportunity. But if the answer is no, or at least not right now, you can reiterate your enthusiasm for the role, briefly explain why you can't move at this time, and offer an alternative, like working remotely or out of a local office. Sometimes it's not as clear-cut, and that's OK. You can say you prefer to stay put for xyz reasons, but would be willing to consider relocating for the right opportunity.

## Possible answer to "Are you willing to relocate?"

*"I do love living in Raleigh and would prefer to stay here. However, for the right opportunity I'd be willing to consider relocating if necessary."*

**Read More:** [The Best Responses to "Are You Willing to Relocate?" Depending on Your Situation](#)

## 49. How many tennis balls can you fit into a limousine?

1,000? 10,000? 100,000? Seriously? Well, seriously, you might get asked brain-teaser questions like these, especially in quantitative jobs. But remember that the interviewer doesn't necessarily want an exact number—they want to make sure that you understand what's being asked of you, and that you can set into motion a systematic and logical way to respond. So take a deep breath and start thinking through the math. (Yes, it's OK to ask for a pen and paper!)

**Read More:** [9 Steps to Solving an Impossible Brain Teaser in a Tech Interview \(Without Breaking a Sweat\)](#)

## 50. If you were an animal, which one would you want to be?

Seemingly random personality-test type questions like these come up in interviews because hiring managers want to see how you can think on your feet. There's no wrong answer here, but you'll immediately gain bonus points if your answer helps you share your strengths or personality or connect with the hiring manager. Pro tip: Come up with a stalling tactic to buy yourself some thinking time, such as saying, "Now, that is a great question. I think I would have to say..."

**Read More:** [4 Steps for Answering Off-the-Wall Interview Questions](#)

## 51. Sell me this pen.

If you're interviewing for a sales job, your interviewer might put you on the spot to sell them a pen sitting on the table, or a legal pad, or a water bottle, or just *something*. The main thing they're testing you for? How you handle a high-pressure situation. So try to stay calm and confident and use your body language—making eye contact, sitting up straight, and more—to convey that you can handle this. Make sure you listen, understand your "customer's" needs, get specific about the item's features and benefits, and end strong—as though you were truly closing a deal.

**Read More:** [4 Tips for Responding to "Sell Me This Pen" in an Interview](#)

## 52. Is there anything else you'd like us to know?

Just when you thought you were done, your interviewer asks you this open-ended doozy. Don't panic—it's not a trick question! You can use this as an opportunity to close out the meeting on a high note in one of two ways, Zhang says. First, if there really is something relevant that you haven't had a chance to mention, do it now. Otherwise, you can *briefly* summarize your qualifications. For example, Zhang says, you could say: "I think we've covered most of it, but just to summarize, it sounds like you're looking for someone who can really hit the ground running. And with my previous experience [enumerate experience here], I think I'd be a great fit."

**Read More:** [How to Answer "Is There Anything Else You'd Like Us to Know?"](#)

## 53. Do you have any questions for us?

You probably already know that an interview isn't just a chance for a hiring manager to grill you—it's an opportunity to sniff out whether a job is the right fit from your perspective. What do you want to know about the position? The company? The department? The team? You'll cover a lot of this in the actual interview, so have a few less-common questions ready to go. We especially like questions targeted to the interviewer ("What's your favorite part about working here?") or the company's growth ("What can you tell me about your new products or plans for growth?") If you're interviewing for a remote role, there are some specific [questions](#) you might want to ask related to that.

**Read More:** [57 Smart Questions to Ask in a Job Interview in 2022](#)

## Bonus questions

Looking for more common interview questions and answers examples? Check out these lists of inquiries for different types of jobs.

- [Behavioral interview questions](#)
- [Phone interview questions](#)
- [Remote interview questions](#)
- [Second interview questions](#)

- [COVID-related interview questions](#)
- [Diversity and inclusion interview questions](#)
- [Emotional intelligence interview questions](#)
- [Internship interview questions](#)
- [Manager interview questions](#)
- [Account management interview questions](#)
- [Accounting interview questions](#)
- [Administrative assistant interview questions](#)
- [Brand management interview questions](#)
- [Customer service interview questions](#)
- [Data science interview questions](#)
- [Digital marketing interview questions](#)
- [Financial analyst interview questions](#)
- [IT interview questions](#)
- [Nursing interview questions](#)
- [Product marketing interview questions](#)
- [Project management interview questions](#)
- [Retail interview questions](#)
- [Sales interview questions](#)
- [Software engineering interview questions](#)
- [Teaching interview questions](#)

**Want even *more* advice for answering common interview questions?**

If you are looking for more in depth advice about these 53 interview questions and how to answer them, here's a list of articles with detailed guides to teach you how to approach your responses.

1. [Tell me about yourself.](#)
2. [Walk me through your resume.](#)
3. [How did you hear about this position?](#)
4. [Why do you want to work at this company?](#)
5. [Why do you want this job?](#)
6. [Why should we hire you?](#)
7. [What can you bring to the company?](#)
8. [What are your greatest strengths?](#)
9. [What do you consider to be your weaknesses?](#)
10. [What is your greatest professional achievement?](#)
11. [Tell me about a challenge or conflict you've faced at work, and how you dealt with it.](#)
12. [Tell me about a time you demonstrated leadership skills.](#)
13. [What's a time you disagreed with a decision that was made at work?](#)
14. [Tell me about a time you made a mistake.](#)
15. [Tell me about a time you failed.](#)
16. [Why are you leaving your current job?](#)
17. [Why were you fired?](#)
18. [Why was there a gap in your employment?](#)
19. [Can you explain why you changed career paths?](#)
20. [What's your current salary?](#)
21. [What do you like least about your job?](#)
22. [What are you looking for in a new position?](#)

23. [What type of work environment do you prefer?](#)
24. [What's your work style?](#)
25. [What's your management style?](#)
26. [How would your boss and coworkers describe you?](#)
27. [How do you deal with pressure or stressful situations?](#)
28. [What do you like to do outside of work?](#)
29. [Are you planning on having children?](#)
30. [How do you stay organized?](#)
31. [How do you prioritize your work?](#)
32. [What are you passionate about?](#)
33. [What motivates you?](#)
34. [What are your pet peeves?](#)
35. [How do you like to be managed?](#)
36. [Do you consider yourself successful?](#)
37. [Where do you see yourself in five years?](#)
38. [How do you plan to achieve your career goals?](#)
39. [What are your career aspirations?](#)
40. [What's your dream job?](#)
41. [What other companies are you interviewing with?](#)
42. [What makes you unique?](#)
43. [What should I know that's not on your resume?](#)
44. [What would your first few months look like in this role?](#)
45. [What are your salary expectations?](#)
46. [What do you think we could do better or differently?](#)
47. [When can you start?](#)

48. [Are you willing to relocate?](#)
  49. [How many tennis balls can you fit into a limousine?](#)
  50. [If you were an animal, which one would you want to be?](#)
  51. [Sell me this pen.](#)
  52. [Is there anything else you'd like us to know.](#)
  53. [Do you have any questions for us?](#)
- 



The Muse is a values-based careers site that helps people navigate every aspect of their careers and search for jobs at companies whose people, benefits, and values align with their unique professional needs. The Muse offers [expert advice](#), [job opportunities](#), a peek behind the scenes at [companies hiring now](#), and [career coaching services](#). You can also find The Muse on [Facebook](#), [Twitter](#), [LinkedIn](#), [Instagram](#), [TikTok](#), and [Flipboard](#).

MORE FROM THE MUSE EDITORS

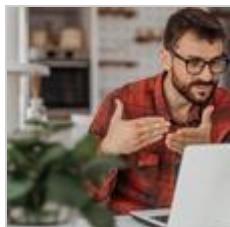
## Sign up for our newsletter

We'll deliver you emails with career advice to guide every stage of your journey and help you find jobs at inspiring companies that are hiring.

SIGN UP

By signing up, you agree to our [Terms of Service](#) & [Privacy Policy](#).

## Recent Articles in Interviewing



REGINA BORSELLINO

**How to Answer "Tell Me About a Time You Made a Mistake" in an Interview**

LILY ZHANG

**4 Better Ways to Answer "Why Should We Hire You?"**

KELLI NEWMAN MASON

**8 Interview Questions About Diversity and Inclusion Every Job Seeker Should Be Able to Answer**

JACLYN WESTLAKE

**How to Do a Practice Interview That'll Actually Help You**

THE MUSE EDITORS

**30+ Best Tips on How to Prepare for a Job Interview**

STAV ZIV

**The Best Ways to Talk About Your Strengths and Weaknesses in a Job Interview**

© 2024 Daily Muse Inc.

**ABOUT THE MUSE**[About Us](#)[Careers](#)[FAQs](#)[Contact Us](#)[Search Jobs](#)**POPULAR JOBS**[New York Jobs](#)[San Francisco Jobs](#)[Seattle Jobs](#)[Engineering Jobs](#)[Marketing Jobs](#)**GET INVOLVED**[For Employers](#)[For Partners](#)[The Muse Book: The New Rules of Work](#)[Podcast: The New Rules of Work](#)

[Browse Companies](#)[Information Technology Jobs](#)[For Career Coaches](#)[Terms of Use](#)[Salaries](#)[Send Feedback About The Muse](#)[Privacy Policy](#)[Search Job Titles](#)[Tell A Friend](#)**JOIN THE CONVERSATION**

**HIRING**

# 27 Most Commonly Asked Job Interview Questions and Answers

Want to know (or use) some of the most commonly asked interview questions and answers? Here's a comprehensive list used by interviewers. ☺

EXPERT OPINION BY JEFF HADEN, CONTRIBUTING EDITOR, INC. @JEFF\_HADEN

JUN 20, 2016

---



Photo: Getty Images

---

While some job interviewers take a fairly unusual approach to interview questions, most job interviews involve an exchange of common interview questions and answers (including some of the most often asked behavioral

interview questions). Here are some of the most common interview questions, along with the best way to answer them.

## 1. "Tell me a little about yourself."

If you're the interviewer, there's a lot you should already know: The candidate's resume and cover letter should tell you plenty, and LinkedIn and Twitter and Facebook and Google can tell you more.

The goal of an interview is to determine whether the candidate will be outstanding in the job, and that means evaluating the skills and attitude required for that job. Does she need to be an empathetic leader? Ask about that. Does she need to take your company public? Ask about that.

If you're the candidate, talk about why you took certain jobs. Explain why you left. Explain why you chose a certain school. Share why you decided to go to grad school. Discuss why you took a year off to backpack through Europe, and what you got out of the experience.

When you answer this question, connect the dots on your resume so the interviewer understands not just what you've done, but also *why*. Interviewers will be listening for a genuine response.

## 2. "What are your biggest weaknesses?"

Every candidate knows how to answer this question: Just pick a theoretical weakness and magically transform that flaw into a strength in disguise.

For example: "My biggest weakness is getting so absorbed in my work that I lose all track of time. Every day I look up and realize everyone has gone home. I know I should be more aware of the clock, but when I love what I'm doing I just can't think of anything else."

So your "biggest weakness" is that you'll put in more hours than everyone else? Great. A seasoned interviewer has a BS detector that can easily pick up on such silliness.

A better approach for a candidate is to choose an actual weakness, but one you're working to improve. Share what you're doing to overcome that weakness. No one is perfect, but showing you're willing to honestly self-assess and then seek ways to improve comes pretty darned close.

### 3. "What are your biggest strengths?"

Although a job candidate's resume and experience should make their strengths readily apparent, interviewers ask this popular interview question because they want to know how their skills can drive a long-term growth in the company.

If you're asked, provide a sharp, on-point answer. Be clear and precise. If you're a great problem solver, don't just say that: Provide a few examples, pertinent to the opening, that *prove* you're a great problem solver. If you're an emotionally intelligent leader, don't just say that: Provide a few examples that prove you know how to answer the unasked question.

In short, don't just claim to have certain attributes—*prove* you have those attributes.

#### **4. "Where do you see yourself in five years?"**

Answers to this question go one of two basic ways. Candidates try to show their incredible ambition (because that's what they think you want) by providing an extremely optimistic answer: "I want your job!" Or they try to show their humility (because that's what they think you want) by providing a meek, self-deprecating answer: "There are so many talented people here. I just want to do a great job and see where my talents take me."

In either case, you learn nothing, other than possibly how well candidates can sell themselves.

For interviewers, here's a better question: "What business would you love to start?"

That question applies to any organization, because every employee at every company should have an entrepreneurial

mind-set.

The business a candidate would love to start tells you about her hopes and dreams, her interests and passions, the work she likes to do, the people she likes to work with -- so just sit back and listen.

## 5. "Out of all the candidates, why should we hire you?"

Since candidates cannot compare themselves with people they don't know, all they can do is describe their incredible passion and desire and commitment and, well, basically beg for the job. (Way too many interviewers ask the question and then sit back, arms folded, as if to say, "Go ahead. I'm listening. Try to convince me.")

And you learn nothing of substance.

Here's a better interview question: "What do you feel I need to know that we haven't discussed?" Or even, "If you could get a do-over on one of my questions, how would you answer it now?"

Rarely do candidates come to the end of an interview feeling they've done their best. Maybe the conversation went in an unexpected direction. Maybe the interviewer focused on one aspect of their skills and totally ignored other key attributes. Or

maybe candidates started the interview nervous and hesitant, and now wish they could go back and better describe their qualifications and experience.

Plus, think of it this way: Your goal as an interviewer is to learn as much as you possibly can about every candidate, so don't you want to give them the chance to ensure you do?

Just make sure to turn this part of the interview into a conversation, not a soliloquy. Don't just passively listen and then say, "Thanks. We'll be in touch." Ask follow-up questions. Ask for examples.

And of course if you're asked this question, use it as a chance to highlight things you haven't been able to touch on.

## **6. "How did you learn about the opening?"**

Job boards, general postings, online listings, job fairs – most people find their first few jobs that way, so that's certainly not a red flag.

But a candidate who continues to find each successive job from general postings probably hasn't figured out what he or she wants to do – and where he or she would like to do it.

He or she is just looking for a job; often, *any* job.

So don't just explain how you heard about the opening. Show that you heard about the job through a colleague, a current employer, by following the company—show that you know about the job *because you want to work there.*

Employers don't want to hire people who just want a job; they want to hire people who want a job with *their* company.

## 7. "Why do you want *this* job?"

Now go deeper. Don't just talk about why the company would be great to work for; talk about how the position is a perfect fit for what you hope to accomplish, both short-term and long-term.

And if you don't know why the position is a perfect fit, look somewhere else. Life is too short. A skilled interviewee loves this interview question because a well-crafted answer shows interviewers the value you will bring to the company.

## 8. "What do you consider your biggest professional achievement?"

Here's an interview question that definitely requires an answer relevant to the job. If you say your biggest achievement was improving throughput by 18 percent in six months but you're interviewing for a leadership role in human resources, that answer is interesting but ultimately irrelevant.

Instead, talk about an underperforming employee you "rescued," or how you overcame infighting between departments, or how so many of your direct reports have been promoted.

The goal is to share achievements that let the interviewer imagine you in the position – and see you succeeding.

## **9. "Tell me about the last time a co-worker or customer got angry with you. What happened?"**

Conflict is inevitable when a company works hard to get things done. Mistakes happen. Sure, strengths come to the fore, but weaknesses also rear their heads. And that's OK. No one is perfect.

But a person who tends to push the blame – and the responsibility for rectifying the situation – onto someone else is a candidate to avoid. Hiring managers would much rather choose candidates who focus not on blame but on addressing and fixing the problem.

Every business needs employees who willingly admit when they are wrong, step up to take ownership for fixing the problem, and, most important, learn from the experience.

## **10. "Describe your dream job."**

Three words describe how you should answer this question: relevance, relevance, relevance.

But that doesn't mean you have to make up an answer. You can learn something from every job. You can develop skills in every job. Work backward: Identify things about the job you're interviewing for that will help you if you do land your dream job someday, and then describe how those things apply to what you hope to someday do.

And don't be afraid to admit that you might someday move on, whether to join another company or – better – to start your own business. Employers no longer expect "forever" employees.

## 11. "Why do you want to leave your current job?"

Let's start with what you *shouldn't* say (or, if you're the interviewer, what are definite red flags).

Don't talk about how your boss is difficult. Don't talk about how you can't get along with other employees. Don't bad-mouth your company.

Instead, focus on the positives a move will bring. Talk about what you want to achieve. Talk about what you want to learn. Talk about ways you want to grow, about things you want to

accomplish; explain how a move will be great for you *and* for your new company.

Complaining about your current employer is a little like people who gossip: If you're willing to speak badly of someone else, you'll probably do the same to me. An experienced interviewer will push to get a real answer by asking this follow-up question: How is your current employer not satisfying these needs?

## 12. "What kind of work environment do you like best?"

Maybe you love working alone, but if the job you're interviewing for is in a call center, that answer will do you no good.

So take a step back and think about the job you're applying for and the company's culture (because every company has one, whether intentional or unintentional). If a flexible schedule is important to you, but the company doesn't offer one, focus on something else. If you like constant direction and support and the company expects employees to self-manage, focus on something else.

Find ways to highlight how the company's environment will work well for you – and if you can't find ways, don't take the job, because you'll be miserable.

When you're the one in the interviewer's seat, you'll get a more honest answer by asking: What do you care about most at work?

### **13. "Tell me about the toughest decision you had to make in the past six months."**

The goal of this question is to evaluate the candidate's reasoning ability, problem-solving skills, judgment, and possibly even willingness to take intelligent risks.

Having no answer is a definite warning sign. *Everyone* makes tough decisions, regardless of their position. My daughter worked part-time as a server at a local restaurant and made difficult decisions all the time – like the best way to deal with a regular customer whose behavior constituted borderline harassment.

A good answer proves you can make a difficult analytical or reasoning-based decision – for example, wading through reams of data to determine the best solution to a problem.

A great answer proves you can make a difficult interpersonal decision, or better yet a difficult data-driven decision that includes interpersonal considerations and ramifications.

Making decisions based on data is important, but almost every decision has an impact on people as well. The best candidates

naturally weigh all sides of an issue, not just the business or human side exclusively.

## 14. "What is your leadership style?"

This is a tough question to answer without dipping into platitudes. Try sharing leadership examples instead. Say, "The best way for me to answer that is to give you a few examples of leadership challenges I've faced," and then share situations where you dealt with a problem, motivated a team, worked through a crisis. Explain *what* you did and that will give the interviewer a great sense of how you lead.

And, of course, it lets you highlight a few of your successes.

## 15. "Tell me about a time you disagreed with a decision. What did you do?"

No one agrees with every decision. Disagreements are fine; it's what you do when you disagree that matters. (We all know people who love to have the "meeting after the meeting," where they've supported a decision in the meeting but then go out and undermine it.)

Show that you were professional. Show that you raised your concerns in a productive way. If you have an example that proves you can effect change, great – and if you don't, show that you

can support a decision even though you think it's wrong (as long as it's not unethical, immoral, etc.).

Every company wants employees to be honest and forthright, to share concerns and issues, but to also get behind a decision and support it as if they agreed, even if they didn't.

## **16. "Tell me how you think other people would describe you."**

I hate this question. It's a total throwaway. But I did ask it once, and got an answer I really liked.

"I think people would say that what you see is what you get," the candidate said. "If I say I will do something, I do it. If I say I will help, I help. I'm not sure that everyone likes me, but they all know they can count on what I say and how hard I work."

Can't beat that.

## **17. "What can we expect from you in your first three months?"**

Ideally the answer to this should come from the employer: They should have plans and expectations for you.

But if you're asked, use this general framework:

You'll work hard to determine how your job creates value – you won't just stay busy, you'll stay busy doing the right things.

You'll learn how to serve all your constituents – your boss, your employees, your peers, your customers, and your suppliers and vendors.

You'll focus on doing what you do best – you'll be hired because you bring certain skills, and you'll apply those skills to make things happen.

You'll make a difference – with customers, with other employees, to bring enthusiasm and focus and a sense of commitment and teamwork.

Then just layer in specifics that are applicable to you and the job.

## **18. "What do you like to do outside of work?"**

Many companies feel cultural fit is extremely important, and they use outside interests as a way to determine how you will fit into a team.

Even so, don't be tempted to fib and claim to enjoy hobbies you don't. Focus on activities that indicate some sort of growth: skills you're trying to learn, goals you're trying to accomplish. Weave those in with personal details. For example, "I'm raising a family,

so a lot of my time is focused on that, but I'm using my commute time to learn Spanish."

## 19. "What was your salary in your last job?"

While this question has been a long-time favorite, new laws now make asking about salary history off limits in some cities and states, including California, Massachusetts, and New York. The aim of the new prohibitions is to end the cycle of pay discrimination.

Even so, it's still OK to ask: Does your current salary really reflect the market rate in your field? Or, is your current salary what you hope to get in your next job?

For job candidates, this is a tough one. You want to be open and honest, but frankly, some companies ask the question as the opening move in salary negotiations. Try an approach recommended by Liz Ryan. When asked, say, "I'm focusing on jobs in the \$50K range. Is this position in that range?" (Frankly, you should already know—but this is a good way to deflect.)

Maybe the interviewer will answer; maybe she won't. If she presses you for an answer, you'll have to decide whether you want to share or demur. Ultimately your answer won't matter too much, because you'll either accept the salary offered or you won't, depending on what you think is fair.

## 20. "A snail is at the bottom of a 30-foot well. Each day he climbs up three feet, but at night he slips back two feet. How many days will it take him to climb out of the well?"

Questions like these are now among the most popular interview questions (thanks, Google) in recent years. As the interviewer, you're not necessarily looking for the right answer, but instead a little insight into the job candidate's reasoning abilities.

Listen carefully as they talk through their logic as they try to solve the problem. Who cares if people get it wrong? The best candidates will laugh at themselves. Assess how the interviewee deals with failure. Moreover, do they know how to be a good communicator?

## 21. "What questions do you have for me?"

A great candidate doesn't waste this opportunity. Smart questions not only signal a great candidate but also reveal to the candidate whether the company is a good fit for them. After all, they're being interviewed, but they're also interviewing the company and have a keen sense of the questions to ask during an interview.

Here goes:

## 22. "What do you expect me to accomplish in the first 90 days?"

A great job candidate will ask this question if they haven't already gleaned this information at some point in their interview. Why? High performers want to hit the ground running. They don't want to spend weeks or months "getting to know the organization." They don't want to spend huge chunks of time in orientation, in training, or in the futile pursuit of getting their feet wet.

They want to make a difference--and they want to make that difference *right now*.

## 23. "What are the three traits your top performers have in common?"

Great candidates also want to be great employees. They know every organization is different – and so are the key qualities of top performers in those organizations. Maybe your top performers work longer hours. Maybe creativity is more important than methodology. Maybe constantly landing new customers in new markets is more important than building long-term customer relationships. Maybe the key is a willingness to spend the same amount of time educating an entry-level customer as helping an enthusiast who wants high-end equipment.

Great candidates want to know, because:

they want to know if they will fit in, and,  
if they do fit in, they want to know how they can be a  
top performer.

## 24. "What *really* drives results in this job?"

Employees are investments, and you expect every employee to generate a positive return on his or her salary. (Otherwise why do you have them on the payroll?)

In every job, some activities make a bigger difference than others. You need your HR team to fill job openings, but what you really want is for them to find the right candidates, because that results in higher retention rates, lower training costs, and better overall productivity.

You need your service techs to perform effective repairs, but what you really want is for those techs to identify ways to solve problems and provide other benefits – in short, to build customer relationships and even generate additional sales.

Great candidates want to know what truly makes a difference and drives results, because they know helping the company succeed means they will succeed as well.

## 25. "What are the company's highest-priority goals this year, and how would my role contribute?"

Is the job the candidate will fill important? Does that job *matter*?

Great candidates want a job with meaning, with a larger purpose – and they want to work with people who approach their jobs the same way.

Otherwise a job is just a job.

## 26. "What percentage of employees was brought in by current employees?"

Employees who love their jobs naturally recommend their company to their friends and peers. The same is true for people in leadership positions – people naturally try to bring on board talented people they previously worked with. They've built relationships, developed trust, and shown a level of competence that made someone go out of their way to follow them to a new organization.

And all of that speaks incredibly well to the quality of the workplace and the culture.

## 27. "What do you plan to do if ...?"

Every business faces a major challenge: technological changes, competitors entering the market, shifting economic trends. There's rarely one of Warren Buffett's moats protecting a small business.

So while some candidates may see your company as a stepping-stone, they still hope for growth and advancement. If they do eventually leave, they want it to be on their terms, not because you were forced out of business.

Say I'm interviewing for a position at your ski shop. Another store is opening less than a mile away: How do you plan to deal with the competition? Or you run a poultry farm (a huge industry in my area): What will you do to deal with rising feed costs?

Great candidates don't just want to know what you think; they want to know what you plan to do – and how they will fit into those plans.



Now accepting applications for Inc.'s Best Workplace awards. [Apply by February 16](#) for your chance to be featured!

Like this column? Sign up to [subscribe to email alerts](#) and you'll never miss a post.

*The opinions expressed here by Inc.com columnists are their own, not those of Inc.com.*

## Modern CEO

from the publisher of *Fast Company* and *Inc.*

A refreshed look at leadership from  
the desk of CEO and chief content  
officer Stephanie Mehta

Email \*

[Sign Up](#)

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

[PRIVACY POLICY](#)

---

**SPONSORED BUSINESS CONTENT**

[PRIVACY POLICY](#)  
[NOTICE OF COLLECTION](#)  
[DO NOT SELL MY DATA](#)  
[AD VENDOR POLICY](#)  
[TERMS OF USE](#)  
[ADVERTISE](#)  
[HELP CENTER](#)  
[ABOUT US](#)  
[SITEMAP](#)

COPYRIGHT 2024 MANSUETO VENTURES 

Inc.com adheres to NewsGuard's nine standards of credibility and transparency.

[LEARN MORE](#)