



InterviewBit

EPAM Interview Questions



To view the live version of the page, [click here](#).

© Copyright by Interviewbit

Contents

EPAM Recruitment Process

1. Interview Process
2. Interview Rounds

EPAM Technical Interview Questions: Freshers and Experienced

3. What is the purpose of the final, finally, and finalize keyword?
4. State the difference between HashMap and Hashtable.
5. Explain OOP concepts.
6. Explain abstract class and abstract method.
7. Write a program to implement abstraction.
8. Can you explain the difference between ArrayList and Vector?
9. Is Java a pass-by-value or reference? Prove it.
10. Could you tell me how many data structures you are familiar with?
11. Write an algorithm for quicksort.
12. What is the best case and worst case time complexity for the quick sort?.
13. Write the binary search algorithm?
14. State the difference between quick sort and merge sort.
15. What is the requirement traceability matrix?
16. What happens when a method or variable contains the static keyword?
17. Could you explain how the stack can be implemented using the array?
18. What are constructors? Write its types.

EPAM Technical Interview Questions: Freshers and Experienced

(.....Continued)

19. Why does Java not support multiple inheritances?
20. Write a program to count the number of vowels occurring in all the substrings of a given String.
21. Can you explain each keyword from the public static void main(String args[])>?
22. Explain Java Collections.

EPAM Interview Preparation

23. Interview Preparation Tips

Frequently Asked Questions

24. Why Epam?
25. What is the Epam test?
26. Is the Epam interview difficult?
27. How do I get placed in EPAM?
28. What is the salary for freshers in Epam?
29. Is Epam only for Java?
30. How long does the Epam interview process take?
31. What is the Eligibility Criteria at Epam?

Let's get Started

Since 1993, EPAM Systems has harnessed its advanced software engineering heritage to become the foremost global provider of digital transformation services - leading the industry in both physical and digital product development and platform engineering services. For the last three years in a row, EPAM has been ranked as the top IT services company on Fortune's Fastest-Growing Companies list.

EPAM offers global standard processes, Leadership Development, Training, and personalized career advancement. Epam employees enjoy a competitive salary, flexibility in their work-life balance, health coverage, ongoing training programs, relocation options, and opportunities for recognition for their open source contributions. EPAM will keep you on your toes and push you to grow. You can learn from seniors and contribute to a global work environment. If you are a tech-savvy individual, you may consider building a long career in the company. Surely, you are wondering how and what you need to prepare for your Epam interview?



In this article, the InterviewBit team has compiled an exhaustive list of EPAM Interview Questions and Answers that cater to both Freshers and Experienced Professionals. Also included are tips on how to prepare for EPAM's interview, as well as information about EPAM's recruitment process. Review them and fine-tune your responses before your EPAM interview.

Now, let's take a look at the EPAM Recruitment Process.

EPAM Recruitment Process

1. Interview Process

EPAM Recruitment is most notable for its unique, yet practical selection process. The company believes that inclusive leadership is built by combining multiple perspectives and experiences, and fair hiring is the first step toward achieving this. With their unique recruitment process, EPAM judges applicants based on the perks that are applicable in real life. Recruiters will assess a candidate's practical knowledge of core subjects, programming skills, and verbal communication skills. Check out the EPAM Recruitment process below to learn how the recruitment team will hone in on the hiring process. Applicants will undergo five rounds of interviews assessing their technical and analytical abilities.

- **Round 1:** Coding assessment - Online
- **Round 2:** Coding assessment - Offline
- **Round 3:** Collaborative Discussion (GD Round) - Offline
- **Round 4:** Technical Interview - Offline
- **Round 5:** HR Interview - Offline

Note: EPAM recruits candidates through both on-campus and off-campus placements. Interested applicants can reach out to the organization either by attending an on-campus placement drive or by submitting an online application on the organization's official website.

2. Interview Rounds

1. Coding challenge (Core Java) - Online

The 1st round is the Java Online Coding Challenge, where the candidates will have to solve 3 EPAM coding questions that will be related to Core concepts (such as arrays, stacks, queues, linked lists, graphs, etc). The EPAM Test is unique in that it must be solely written in Java only. It is mandatory that candidates use the webcam in the first round; they cannot leave their seats during the test. Take a look at the EPAM test pattern as follows:

Questions	Marks	Time Frame (150 min)
1	20	40 minutes
2	30	50 minutes
3	50	60 minutes

2. Coding challenge (Core Java) - Offline

Candidates who clear the first round of screening will be invited to the second round of offline coding assessment. Generally, this round is comprised of MCQs (pertaining to Java, OOPS, DBMS, Software Testing, Networking, and Operating Systems) and 2 coding questions that range from medium to hard level. For coding, the questions are based on Core Concepts (Java, Data Structure and Algorithm, Object-Oriented Programming, Dynamic programming, etc). Brush up on these topics before you appear. Take a look at the test pattern as follows:

Sections	Marks	Time Frame
MCQs	20	120 minutes
1st Coding Question	30	
2nd Coding Question	50	

3. Group Discussion

Candidates who meet the qualifying marks will be called for the 3rd round of Group Discussion. In this round, a group of selected candidates will be formed. An instructor will oversee the entire GD round, giving the candidates a topic on which they will be conversing for approximately 10 minutes each. If you want to win this round, just speak up. Essentially, they are testing your English ability and communication skills.

4. Technical Interview Round

Following the GD, the shortlisted candidates will be invited to the 1:1 Epam technical interview. Technical interview questions are designed to assess your technical knowledge and your ability to handle conflict and challenging scenarios related to the position you are interviewing for. Questions will range from technical to problem-solving skills to brain teasers. Thus, you must prepare for every eventuality.

- Interview questions will cover topics such as Java, Object-Oriented Programming (OOP), HTML Code, Database Management Systems (DBMS), Data Structures, and Algorithms (DSA).
- The interviewer will also ask you about your past experiences and projects, how you handled obstacles, and how successful they were.

Note: Don't be afraid to tell the interviewer if you don't know the answer to a specific question (in a polite manner). Passing this round is 90% more likely to lead to a job offer; nevertheless, you must also pass the easiest yet most critical HR interview.

5. HR Interview Round

Having a good academic record and technical knowledge alone will not be sufficient to gain employment. The HR round is usually the final round in a company's recruitment process. Though the HR round may seem easy, it is most critical and essential. At this stage, the hiring manager will look at candidates' personalities, strengths, and weaknesses, along with their communication and reasoning skills. The interviewer will also ask a few behavioural questions to get acquainted with you and determine if your personality and behaviour are a good match for the position. Below you will find some common Epam HR interview questions.

- What does EPAM mean?
- Tell me a little bit about yourself and your family background.
- What makes you confident you are eligible to apply for EPAM?
- How do you envision yourself five years from now?
- How did Epam catch your attention?
- Are you aware of EPAM systems?
- What are your hobbies?
- When a crisis strikes, how well do you cope?
- What makes you a good hire?
- What are your greatest assets and greatest weaknesses?
- Would you like to share your take on the COVID situation??
- Would you like to learn any new technologies in the future besides the ones you already know?
- Have you ever felt like you couldn't overcome a situation? What helped you get through it?
- Would you like to ask us anything?
- Can you manage pressure well?
- What interested you in picking Java as your primary language or is it simply because of EPAM?
- What would you do if a coworker consistently arrived late to a weekly meeting?

EPAM Technical Interview Questions: Freshers and Experienced

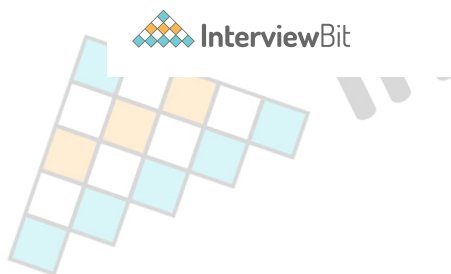
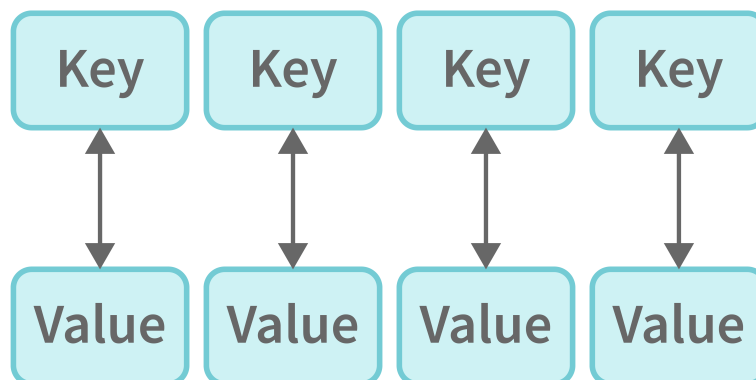
3. What is the purpose of the final, finally, and finalize keyword?

- **Final:** Final is a reserved keyword in Java that can be used in multiple contexts to define an entity. Final variables have fixed values, i.e. they cannot be changed after they have been assigned. Depending on whether it is used for a variable, a class, or a method, the final keyword has different meanings.
 - **Final Variable:** Once a final variable has been initialized, its value cannot be changed.
 - **Final Method:** A subclass cannot override the final method. Declaring a method final means that it cannot be overridden.
 - **Final Class:** When a class is final, it cannot be subclassed. Declaring a class final means that we cannot extend or make a subclass of it in any way.
- **Finally:** Finally is a keyword that is used in conjunction with a try/catch block and ensures that a section of code will be executed, regardless of whether an exception is handled or not. It executes important code such as resource cleanup or releasing memory usage.
- **Finalize:** A garbage collector always calls the finalize method just before it deletes or destroys an object to perform cleanup activities. The clean-up activity means closing resources associated with that object, such as databases and network connections, or de-allocating resources. It is defined in the Object class.

4. State the difference between HashMap and Hashtable.

Both HashMaps and Hashtables store data in the key-value form. Each store's unique keys use hashing techniques.

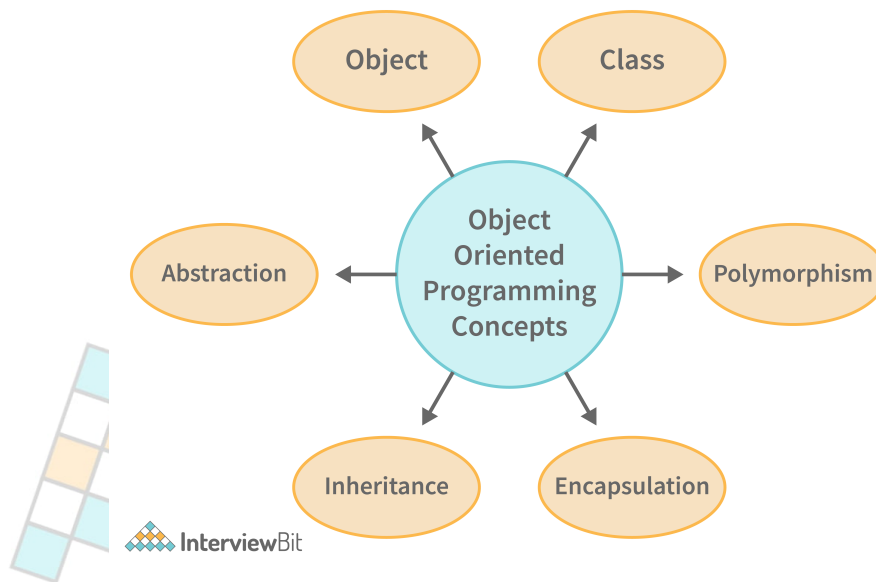
Key-Value Pairs



HashMap	HashTable
It is non-synchronized. As a result, two or more threads can access the HashMap data at the same time.	It is synchronized. As a result, only one thread can access the Hashtable data at a time.
This is not thread-safe, so it cannot be shared among many threads without proper synchronization.	It is thread-safe, so it can be shared among many threads.
In a hashmap, there can be one null key and any number of null values.	Hashtables do not allow null keys or null values.
HashMap values are iterated by using an iterator.	A HashTable, however, also includes an Enumerator along with Iterator for traversing/iterating the value stored in it.
Since HashMaps are unsynchronized, they are faster and use less memory than Hashtables.	Since Hashtables are synchronized, they are slow and use more memory than HashMaps. But, it eliminates the need to write extra code for synchronization.

5. Explain OOP concepts.

The following are some general Java OOP (Object Oriented Programming) concepts:



- **Object:** An Object is an instance of a class, and a program may contain multiple instances of a given class. These objects have physical properties and behaviour.
- **Class:** A class is a group of objects. It is a logical component rather than a physical one. Classes take up no memory and are also called templates for objects.
- **Abstraction:** It refers to the representation of essential features without revealing the details of their implementation. Its primary purpose is to conceal information.
- **Inheritance:** It is a concept of OOP that allows a class (child) to inherit the properties and behaviours of its superclass object (parent). It creates a relationship that resembles that of a parent and a child.
- **Polymorphism:** In Java, this refers to the ability for a variable, object, or function to have multiple forms. Overloading and overriding methods are utilized to achieve polymorphism.
- **Encapsulation:** Essentially, encapsulation is the act of binding together data and code into a single unit. To put it simply, it is a protective shield that prevents data from being accessed by code that exists outside the shield.

6. Explain abstract class and abstract method.

- **Abstract class:** This class can never be instantiated i.e., objects of abstract classes cannot be created. There can be both regular and abstract methods in an abstract class. Abstract classes are declared using the abstract keyword. It is necessary to extend (inherit) the abstract class. The subclass of an abstract class, i.e., the class inherited from it, usually implements all of the abstract methods of the parent class (abstract class).
- **Abstract method:** Abstract methods can only be used in abstract classes, and they do not have bodies. The body of the abstract method is provided by the subclass. The abstract keyword is also used for abstract methods.

Example:

```
// Abstract class
abstract class InterviewBit
{
    // Abstract method (does not have a body)
    abstract void Products();

    // regular method
    void method();
    {
        ...
    }
}
```

7. Write a program to implement abstraction.

Abstraction: Java uses interfaces and abstract classes to implement abstraction. The following program illustrates the implementation of abstraction.

Code:

```
// Java program to illustrate abstraction implementation
//abstract class
abstract class InterviewBit
{
    //abstract method
    abstract void products();
}

//extends the abstract class
class ScalerAcademy extends InterviewBit
{
    @Override
    void products()
    {
        System.out.println("Scaler Academy by InterviewBit");
    }
}

class ScalerEdge extends InterviewBit
{
    @Override
    void products()
    {
        System.out.println("Scale Edge by Scaler");
    }
}

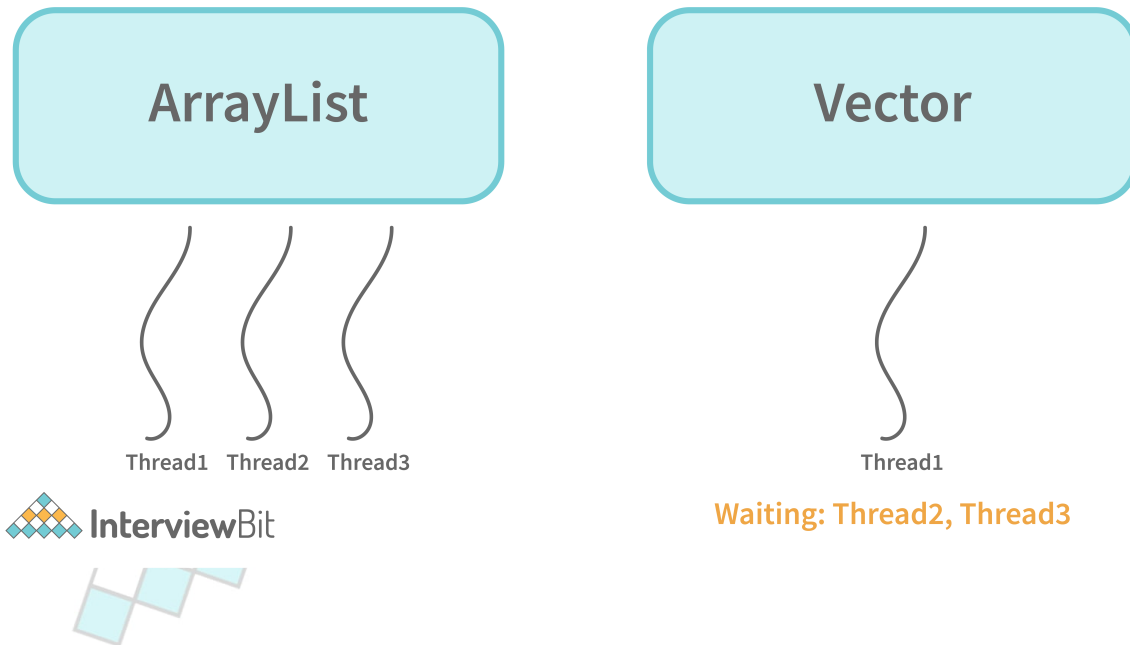
public class Main
{
    public static void main(String[] args)
    {
        InterviewBit obj1 = new ScalerAcademy();
        InterviewBit obj2 = new ScalerEdge();
        //invoking abstract method
        obj1.products();
        obj2.products();
    }
}
```

Output:

```
Scaler Academy by InterviewBit
Scale Edge by Scaler
```

8. Can you explain the difference between ArrayList and Vector?

ArrayList and Vector are both dynamic arrays (resizable arrays) used in Java to implement the list interface.



ArrayList	Vector
By default, ArrayList is not synchronized. This means that ArrayList can be used by multiple threads simultaneously.	It is synchronized. This means that Vector can be used by only one thread at a time.
ArrayList increases its size by 50% if the number of elements exceeds its maximum capacity.	Vector increases its size by 100% if the number of elements exceeds its maximum capacity.
Since ArrayLists are unsynchronized, they are faster as compared to vectors.	Since Vectors are unsynchronized, they are slower as compared to ArrayLists.
The iterator interface can be used to traverse the elements of the ArrayList.	Both the Enumeration interface and Iterator interface can be used to traverse the elements of Vector.

9. Is Java a pass-by-value or reference? Prove it.

Java strictly enforces pass-by-value. Passing the parameters by using pass-by-values does not affect/alter the original variable. In the following program, we have initialized a variable called 'x' with some value and used the pass-by-value technique to demonstrate how the value of the variable remains unchanged.

Code:

```
public class Main
{
    public static void main(String[] args)
    {
        //Original value of 'x' will remain unchanged
        // in case of call-by-value
        int x = 5;
        System.out.println( "Value of x before call-by-value: " + x);

        processData(x);
        System.out.println("Value of x after call-by-value: " + x);
    }
    public static void processData(int x)
    {
        x=x+10;
    }
}
```

Output:

```
Value of x before call-by-value: 5
Value of x after call-by-value: 5
```

10. Could you tell me how many data structures you are familiar with?

Data Structures are specialized ways to store and organize data in computer systems in order to perform operations on the data more efficiently. Data structures are used across a wide range of fields in computer science and software engineering. You should choose the appropriate data structure based on your requirements and project. In the case of storing data sequentially in memory, the Array data structure is recommended. In general, data structures can be divided into two categories:

Linear data structure: In the linear data structure, elements are arranged one after another in sequence. As the elements are organized in a specific order, they are easily implemented. A few examples of linear data structures are as follows:

- **Array:** In an array, elements are sequentially stored one after another in memory. They all have the same data type.
- **Stack:** Data elements in the stack are arranged based on the LIFO (Last in First Out) principle. In other words, the last element stored in a stack will be removed first.
- **Queue:** Data elements in Queue are arranged based on the FIFO (First in First out) principle. In other words, the first element stored in a queue will be removed first.
- **Linked List:** The elements of a linked list are stored as nodes. Each node contains the data items and the address to the next node. Generally, a node is made up of two parts, one storing the data element, and the other storing the link to the next node.

Non-linear data structure: In non-linear data structures, elements are not in any particular order. Rather, they are organized in a hierarchical manner in which one element is linked to another or more elements. A few examples of non-linear data structures are:

- **Graph:** Graphs are collections of edges and vertices. Graphs consist of nodes called vertices, and vertices are connected to each other by edges.
- **Trees:** Trees are also made up of vertices and edges, just like graphs. Tree data structures allow only one edge to exist between two vertices.

11. Write an algorithm for quicksort.

Sorting entails arranging items in a systematic manner. With the QuickSort algorithm, data elements can be sorted much faster than with any other sorting algorithm. This algorithm is based on the divide and conquers approach. It begins by selecting an element from the array known as the pivot element. It then uses the partition algorithm to divide the array into two sub-arrays, one containing greater values than the pivot element and the other containing smaller values than the pivot element. After that, the quicksort algorithm recursively sorts these two sub-arrays. As soon as the sorting is completed, both sub-arrays are combined into one sorted array.

```
import java.util.Arrays;
class Quicksort
{
    // method to find the partition position
    static int partition(int array[], int leftmost, int rightmost)
    {
        //leftmost element as pivot
        int pivot = array[rightmost];

        // pointer for greater element
        int i = (leftmost - 1);
        // compare each element with pivot
        for (int j = leftmost; j < rightmost; j++)
        {
            if (array[j] <= pivot)
            {
                // If element found is smaller than pivot
                //it will be swapped with the larger element pointed by i
                i++;
                // element at i is swapped with element at j
                int s = array[i];
                array[i] = array[j];
                array[j] = s;
            }
        }
        // swapt the pivot element with the greater element specified by i
        int s = array[i + 1];
        array[i + 1] = array[rightmost];
        array[rightmost] = s;
        // return the position from where partition is done
        return (i + 1);
    }
    static void quickSort(int array[], int leftmost, int rightmost)
    {
        if (leftmost < rightmost)
        {
            int pi = partition(array, leftmost, rightmost);

            // recursive call on the left of pivot
            quickSort(array, leftmost, pi - 1);
            // recursive call on the right of pivot
            quickSort(array, pi + 1, rightmost);
        }
    }
}
class Main
{
    public static void main(String args[])
    {
        int[] arr = { 5, 4, 6, 1, 0, 8, 3 };
        System.out.println("Unsorted Array");
        System.out.println(Arrays.toString(arr));
        int size = arr.length;
        Quicksort.quickSort(arr, 0, size - 1);
        System.out.println("Sorted Array in Ascending Order: ");
        System.out.println(Arrays.toString(arr));
    }
}
```

Output:

```
Unsorted Array  
[5, 4, 6, 1, 0, 8, 3]  
  
Sorted Array in Ascending Order:  
[0, 1, 3, 4, 5, 6, 8]
```

12. What is the best case and worst case time complexity for the quick sort?.

Quicksort can be applied in several different ways based on the position of the pivot in the array:

- Selecting the first or last element in the array as a pivot element
- Selecting a median element of an array as a pivot element

Best Case: Quicksort performs the best if we divide arrays and subarrays into two equal partitions, i.e., when we select the median element of the array as a pivot element. The partition positions for each element happen to be in the middle of the array, resulting in a balanced binary tree of height $\log(n)$. Therefore, the total complexity becomes $O(n \log n)$.

Worst Case: Worst case is when we select the left-most or right-most element of an array as a pivot element. The partition positions for each element happen to be at the edge of the array (left or right), resulting in a skew tree having height n . Therefore, the total complexity becomes $O(n^2)$.

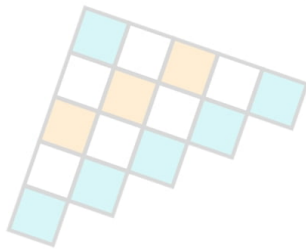
13. Write the binary search algorithm?

A binary search algorithm is used to determine an element's position within a sorted array. This algorithm is based on the divide and conquers approach. A binary search looks for the target element by comparing it with the middlemost element in the array.

Example: Assume that the array to be searched is 1234567. Suppose $x = 5$ is the element to be searched. 1 is the leftmost element (low) and 7 is the rightmost element (high) element in the array. Find the middle element (mid) of the array using the formula $[(low + high)/2]$. In this case, the middle element is $(1+7)/2=4$.

- If $x == mid$, it will return mid.
- If $x > mid$, x will be compared against the middle element of the subarray to the right of mid.
- If $x < mid$, x will be compared against the middle element of the subarray to the left of mid.

Code:



```
// Binary Search in Java
import java.util.Scanner;
public class Main
{
    int binarySearch(int array[], int x, int low, int high)
    {
        // Repeat until the pointers low and high meet each other
        while (low <= high)
        {
            int mid = low + (high - low) / 2;
            if (array[mid] == x)
                return mid;
            if (array[mid] < x)
                low = mid + 1;
            else
                high = mid - 1;
        }

        return -1;
    }
    public static void main(String args[])
    {
        Main obj = new Main();
        int array[] = { 1, 2, 3, 4, 5, 6, 7 };
        int n = array.length;
        Scanner s= new Scanner(System.in);
        System.out.print("Enter a target element: ");
        int x=s.nextInt();

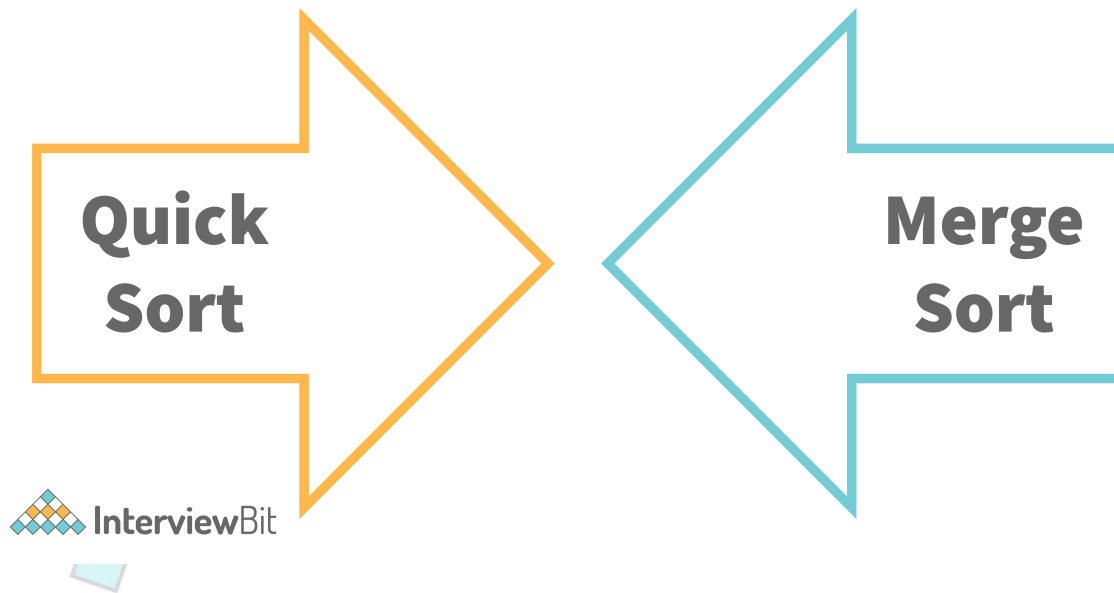
        int result = obj.binarySearch(array, x, 0, n - 1);
        if (result == -1)
            System.out.println("Not found");
        else
            System.out.println("Element found at index " + result);
    }
}
```

Output:

```
Enter a target element: 7
Element found at index 6
```

14. State the difference between quick sort and merge sort.

Both Quicksort and Mergesort are sorting algorithms that allow array elements to be arranged in sequence order, thereby, facilitating a faster and easier search for array elements. Both are based on the divide and conquer approach.



Quicksort	Mergesort
In Quicksort, each element of the array is compared with a pivot element.	Mergesort repeatedly divides the array into two subarrays ($n/2$) until only one element remains.
Quicksort is ideal for small arrays.	Merge sort is suitable for any array size, whether small or large.
The worst time complexity of Quicksort is $O(n^2)$.	The worst time complexity of Mergesort is $O(n \log n)$.
It doesn't take up any more space to perform the quick sort.	To merge two subarrays takes additional space as a temporary array.
For small data sets, it will be faster than other sorting algorithms.	With this algorithm, all data sets are sorted at the same speed.

15. What is the requirement traceability matrix?

As the name suggests, the RTM (Requirement Traceability Matrix) is a document, in the form of a table, which connects and traces user requirements with test cases to make sure that all requirements are covered during testing. RTM was designed in order to verify that all test cases had been covered and that no functionality had been missed during testing. In addition, it assists with identifying and maintaining project requirements and deliverables. Simply put, it is a way of tracking project requirements and ensuring they have been met.

Advantages:

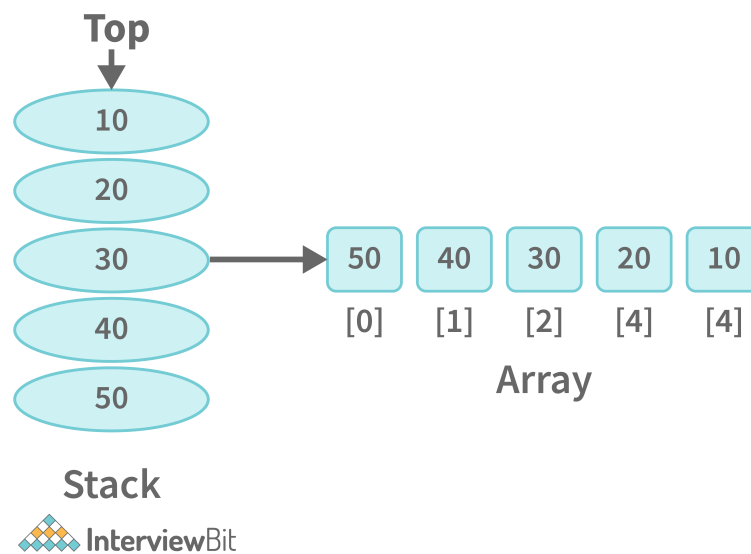
- Ensure 100% test coverage.
- It makes it easy to identify missing functionality.
- Aids you in finding defects related to vital requirements, along with defect severity and priority.
- When requirements change, it is possible to identify test cases that should be updated.
- The overall status of test execution can be tracked easily.

16. What happens when a method or variable contains the static keyword?

Static keywords are mainly used for memory management in Java, and can be used with blocks, methods, variables, and nested classes. When you use a static keyword with a method or variable within a class, those static members will remain constant (you cannot change or modify them) for every instance you create of that particular class.

17. Could you explain how the stack can be implemented using the array?

Stack is a type of linear data structure that follows the LIFO (Last In First Out) principle. Insert and delete operations on the stack are performed from one end (top) only. Inserting a data element on top of the stack is called a push operation, and removing a data element from the top is called a pop operation.



Implementation of stack using Array

- **Using push():** To insert data into the stack, we use the Push() function, but before we do so, we have to check if the stack is full. The push operation cannot be performed if the stack is full, a condition that is sometimes called stack overflow. But, if there is space, the push() function inserts the element and increments the top pointer.

```
//Program to implement stack using Array
//x is element to be added and n is size of the stack
void push (int x, int n)
{
    if (top == n )
        printf("Overflow");
    else
    {
        top = top +1;
        stack[top] = x;
    }
}
```

- **Using pop():** To delete data from the stack, we use the Pop() function, but before we do so, we have to check if the stack is empty. If the stack is empty, the push() operation cannot be performed; otherwise, the pop() function decrements the value at the top of the stack.

```
//Program to implement stack using Array
int pop ()
{
    if(top == -1)
    {
        printf("Underflow");
        return 0;
    }
    else
    {
        return stack[top - - ];
    }
}
```

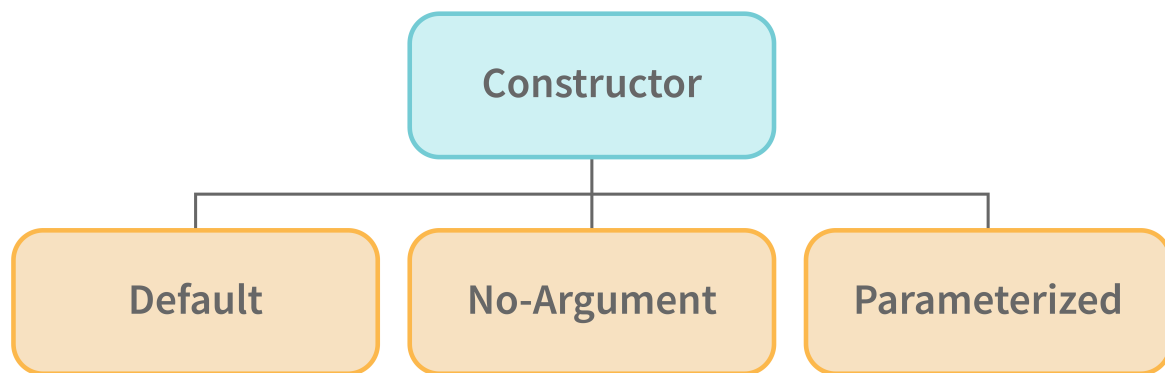
18. What are constructors? Write its types.

Constructors can be defined as a special member function of a class that is responsible for initializing a newly created object of the class. Constructors are automatically called when we create an instance (object) of a class. Unlike other functions, constructors have no return type, and their name should be the same as that of the class they are declared in.

Syntax:

```
class Scaler
{
    Scaler() {
        // constructor body
    }
}
```

Types of Constructors:

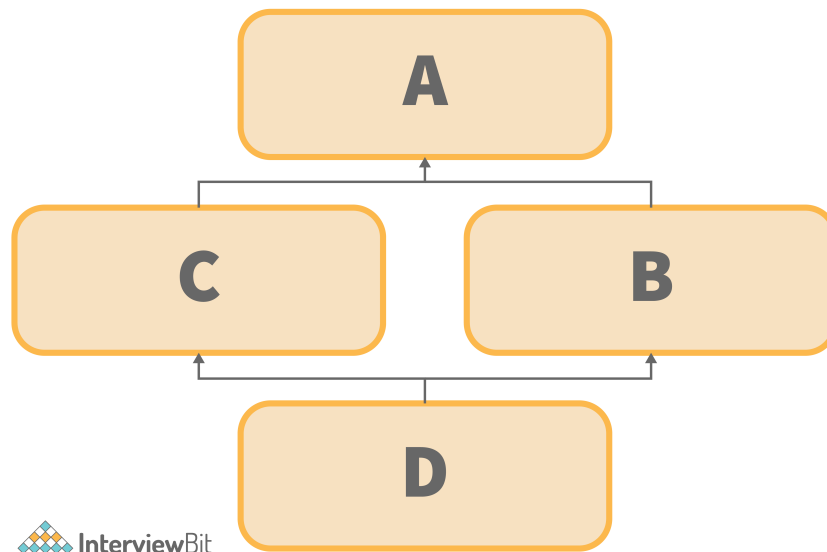


InterviewBit

- **Default Constructors:** When there is no constructor defined, the Java compiler automatically creates a no-arg constructor at runtime, known as the default constructor.
- **No-Arg Constructors:** Like methods, a Java constructor may or may not contain any arguments (parameters). Constructors without any parameters are known as no-argument constructors.
- **Parameterized Constructors:** Constructors with one or more parameters are known as parameterized constructors.

19. Why does Java not support multiple inheritances?

The idea behind multiple inheritances is to combine the behaviour and properties of multiple classes into a single class. Many popular languages support multiple inheritances, including C++ and Common Lisp. Multiple inheritances are not supported in Java due to the ambiguity caused by it. As a result of multiple inheritances, the Diamond problem is a common occurrence.



As you can see, there are two classes B and C that inherit from class A. Suppose, classes B and C override the method `show()` (inherited from class A) and provide their own implementation. Class D inherits properties from both classes B and C, resulting in multiple inheritances. Since class D wants to override the `show()` method, what overridden method would be used? Is it from class B or C? Here we are faced with ambiguity.

20. Write a program to count the number of vowels occurring in all the substrings of a given String.

Consider a string of lowercase characters that contains 0 or more vowels. The string is of length n and can be divided into $n(n+1)/2$ substrings. The simple solution is to count the number of vowels within each substring and then add them together to get the final result. Here is the code to count the number of vowels in all the substrings of a given string.

Code:

```
import java.io.*;
import java.util.*;
import java.util.Scanner;
public class Main {

    //calculates total sum of all vowel occurrences
    static int vowel_calc(String str)
    {
        int n = str.length();
        int arr[] = new int[n];

        for (int i = 0; i < n; i++)
        {
            if (i == 0)
                // Number of times the 0th character occurs in all substrings
                arr[i] = n;
            else
                // Number of times the ith character occurs in all substrings
                arr[i] = (n - i) + arr[i - 1] - i;
        }

        int sum = 0;
        for (int i = 0; i < n; i++) {
            char ch = str.charAt(i);
            // Check to see if ith character is a vowel
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                sum += arr[i];
        }

        // Return total sum of all vowel occurrences
        return sum;
    }

    // Driver Code
    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter a String");
        String str=sc.nextLine();

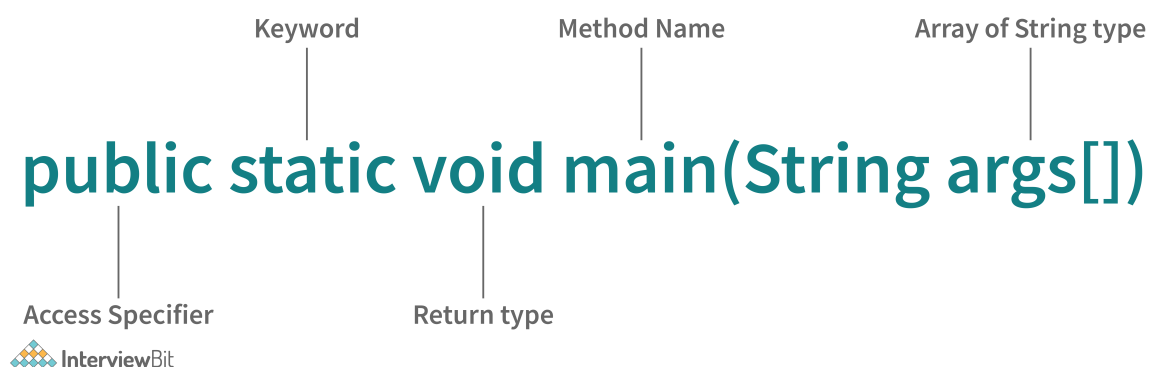
        System.out.println(vowel_calc(str));
    }
}
```

Output:

Enter a String
Scaler
22

21. Can you explain each keyword from the public static void main(String args[])>?

Java's most important method is public static void main(String[] args). Java programs begin with the main method whose syntax is public static void main(String[] args). Only the String array argument name can be changed, for example, args can be changed to myStringArgs.

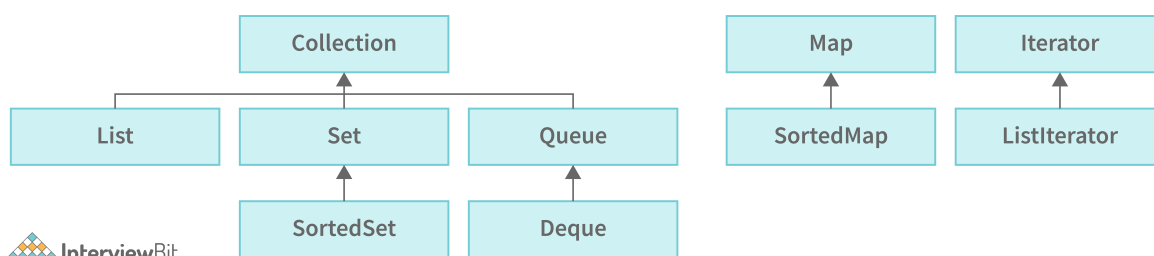


- **Public:** It is an Access modifier that specifies where and who can access the method. When you make a method non-public, it will not be executed by any program, since there are certain restrictions on access.
- **Static:** At the beginning of the Java runtime, there are no objects of the class defined. As `main()` is static, the JVM (Java Virtual Machine) can invoke it without having to instantiate (without creating the object) the class. Furthermore, this prevents the unnecessary consumption of memory by objects that are declared only for calling `main()` methods by the JVM.
- **Void:** Every method in Java must provide a return type. Because Java's main method returns nothing, its return type is void. A Java program terminates once the `main()` method has terminated. Therefore, the `main()` method cannot return any value, as the JVM cannot do anything with its return value.
- **Main:** It's the name of Java's main method. The JVM searches for this identifier as the starting point for the Java program.
- **String[] args:** In Java, the main method takes only one argument of the type String array. These arguments are also known as Java command line arguments.

22. Explain Java Collections.

Java has a separate framework called "Collection Framework" which provides a set of classes and interfaces to implement commonly reusable data structures and algorithms. This framework functions more like a library than a framework. The collections framework usually provides interfaces for various collections and classes for implementing them.

Java Collections Framework



Among the benefits are:

- Java collections framework provides a number of data structures and algorithms that are ready to use. We no longer need to write code to implement these data structures and algorithms.
- The code will be much more efficient since the collections framework has been optimized.
- Collections frameworks also let us use a particular data structure for the specific data type.

Useful Interview Preparation Resources

- [Java Interview Questions](#)
- [Automation Testing Interview Questions](#)
- [Technical Interview Questions](#)
- [Coding Interview Questions](#)
- [Free Mock Interview](#)
- [Interview Guide](#)

EPAM Interview Preparation

23. Interview Preparation Tips

IT interview preparation entails much more than simply looking up frequently asked interview questions. The IT interviewing process is extensive and rigorous, involving the evaluation of your programming abilities, analytical skills, and personal traits. Interviewers put candidates in hypothetical situations to evaluate how well they handle pressures and solve problems. Still, the question remains; How can you persuade a gaggle of potential employers that you are the right person for the job and will not disappoint them? Hence, we thought it would be helpful to compile a list of the top pre-interview tips. Whether you are a tech novice or a pro, here are some tips for acing your next EPAM interview.

- **Perfect your skills:** Practice different types of job interviews, such as the screening interview, group interview, technical interview, the behavioural interview (competency-based). Review past interviews to gain a better understanding of how interviews are conducted, what you can expect, and how you should prepare for them. Search and solve previous years' Question Papers to get a better understanding of the topics and test your knowledge as well as speed and accuracy.
- **Focus on the basics:** Review the job description to figure out what kind of position you're applying for. Research about the organization, look for out what the organization needs, how your background matches, and what you can bring to the organization. Interviewers will primarily ask questions about fundamentals; DSA, operating systems, networking, DBMS, CS concepts, class design, programming language, etc. Fundamental knowledge will equip you to solve open-ended problems - the kind of problems many organizations face every day.
- **Get geeky:** While you are engrossed in a technical conversation, feel free to share your interests and opinions. A room filled with like-minded people naturally fosters conversations around mutually engaging technical topics. Don't miss the shot! Remain enthusiastic and positive throughout the interview.
- **Think of different ways to solve problems:** Interviewers will try to figure out how you will utilize your fundamentals in practice by asking open-ended questions that can be solved in different ways. You will be evaluated on how you articulate open-ended questions, how you choose a solution, and how you communicate effectively. Practice solving questions with the various tools from your toolbox in order to develop a sense of how and when to utilize each of them.
- **Express your thoughts:** Talk through your alternative solutions before you decide which to use when solving a given problem. Doing so will enable the interviewer to see and understand how you approach the problem. It is important to ask clarifying questions and not to enlarge the scope of the problem. Also, ensure that you verbally run through test cases prior to concluding that you have completed the task, as this will allow you to verify the code and demonstrate to the interviewer that you are checking your code.
- **Never jump right in:** Don't jump right in when facing a coding challenge; instead, ask questions first (even if you know the answer already). If you run into trouble, don't be afraid to ask for help. Asking thoughtful questions in your interview can reinforce your interest in the job. After each interview round, do

Frequently Asked Questions

24. Why Epam?

EPAM is an American company that specializes in digital platform engineering and product design, product development, and consulting. They have an excellent work culture, no-frills, excellent digital engineering teams, great leaders, very down-to-earth, and fewer politics. EPAM team members exhibit outstanding teamwork and collaboration. They place more emphasis on quality than quantity. EPAM has many high-quality projects with the newest technologies, so there is a greater chance to work on varied and excellent projects.

25. What is the Epam test?

The EPAM Systems recruitment process consists of five stages; the first of which is an online test. It is a Java Online Coding Challenge, in which the candidates must answer three questions. One unique aspect of the EPAM Test is that the tests are in Java language only.

26. Is the Epam interview difficult?

The EPAM Systems recruitment process usually involves five rounds; the difficulty level ranges from easy to medium. An individual who has good Java programming skills and computer science knowledge can crack the interview. As part of their recruitment process, EPAM judges a candidate solely on the perks that can be expected in real life. You need to be prepared to appear relaxed, calm, and collected during your interview, which is what employers are looking for in their ideal candidate.

27. How do I get placed in EPAM?

EPAM Systems hires graduates for Junior Software Engineer positions through both on-campus and off-campus placements. Interested applicants can reach out to the organization either by attending an on-campus placement drive or by submitting an online application on the organization's official website. Below is a small overview of how to join EPAM.

- Apply for the position that best matches your skills and interests on the company's official website.
- Usually, within two to three business days (and sometimes up to a week), they will review your application and check if you are a good fit for the position. In case the match is right, they will contact you for further processing.
- You will now go through the standard EPAM interview process.
- After the interview, if the feedback is positive, you will receive an offer.

28. What is the salary for freshers in Epam?

Epam package for freshers (Junior Software Engineer) starts at ₹ 3 LPA and goes up to ₹ 11 LPA. An entry-level Software Engineer with less than one year of experience is expected to earn an average salary of ₹ 6 LPA in India.

29. Is Epam only for Java?

As a product-based company, EPM excels in Java development. In addition, the EPAM Systems Recruitment Test is conducted only in JAVA. It is not permissible to use any other language. Even so, this does not imply that you will not be able to work for the organization if you are a Python expert or React.js expert, or any other. They have diverse employment opportunities across multiple locations.

30. How long does the Epam interview process take?

A total of 5 rounds are involved in the Epam hiring process; Java online coding challenge, MCQs, Group Discussion, Technical Interview, and lastly, HR Interview. Epam officials generally follow this interview process; however, they may conduct more rounds depending on the company's requirements, your experience, and the job profile you hold. An individual who has cleared all rounds should receive an offer letter within approximately 5–7 working days.

31. What is the Eligibility Criteria at Epam?

The following are the eligibility criteria for Junior Software Engineer at Epam:

Eligibility Criteria	Details
Graduation	Eligible: B.E/ B.Tech/ M.E/ M.Tech/ MCA
Stream	CSE/ECE/IT/other CS Stream Only
Academic and Graduation Criteria	60% or 6.0 CGPA across 10th, 12th, & UG/PG
Backlogs	No active/current Backlogs



Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)