

## Java Interview Questions and Answers

### ✓What is Java?

Java is a high-level programming language that was developed by James Gosling at Sun Microsystems in the mid-1990s. It is a platform-independent language that can be used to develop applications for desktop, mobile, and web platforms.

### ✓What are the features of Java?

Java has several features that make it a popular programming language, including Platform Independence, Object-Oriented, Simple, Secure, Robust, Multithreaded, Architecture Neutral, Portable, High Performance, Distributed, Dynamic, automatic memory management, multithreading, and robustness.

### ✓What is a class in Java?

A class is a blueprint or prototype from which objects are created. It defines a datatype by bundling data and methods that work on the data into one single unit. A class in Java is a blueprint which includes all your data. A class contains fields (variables) and methods to describe the behavior of an object. Let's have a look at the syntax of a class.

```
class Abc {  
    member variables // class body  
    methods}
```

### ✓What is an object in Java?

An object in Java is an instance of a class that contains data and behavior. It is created from a class using the "new" keyword.

An object has three characteristics:

State

Behavior

Identity

An object is created using the 'new' keyword. For example:

```
ClassName obj = new ClassName();
```

### ✓What is the difference between a class and an object?

A class is a blueprint or template for creating objects, while an object is an instance of a class.

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

In Java, we declared the main function as a public static void main (String args[]).

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.

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.

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.

main: main represents that the function declared is the main function. It helps JVM to identify that the declared function is the main function.

String args[]: It stores Java command-line arguments and is an array of type java.lang.String class.

### ✓What is System.out 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. It will print to the standard out of the system. It is mostly used to display results on the console. It gives output on the console with the default(black) color. 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.

✓What is the platform?

A platform is the hardware or software environment in which a piece of software is executed. There are two types of platforms, software-based and hardware-based. Java provides the software-based platform.

✓Why is the main method static?

Because the object is not required to call the static method. If we make the main method non-static, JVM will have to create its object first and then call main() method which will lead to the extra memory allocation.

✓Can we execute a program without main() method?

No, It was possible before JDK 1.7 using the static block. Since JDK 1.7, it is not possible.

✓Why Java is not 100% Object-oriented?

Java is not 100% Object-oriented because it makes use of eight primitive data types such as boolean, byte, char, int, float, double, long, short which are not objects.

✓Name the Java IDE's?

Eclipse and NetBeans are the IDE's of JAVA.

✓What are JAR files?

Java Archive(JAR) is a file format that aggregates many files into a single file similar in a format similar to a ZIP file. In other words, the JAR is a zipped file comprised of java class files, metadata, and resources like text, images, audio files, directories, .class files, and more.

Security – we can digitally sign JAR files

Compression – while using a JAR, we can compress files for efficient storage

Portability – we can use the same JAR file across multiple platforms

Versioning – JAR files can hold metadata about the files they contain

Sealing – we can seal a package within a JAR file. This means that all classes from one package must be included in the same JAR file

Extensions – we can use the JAR file format to package modules or extensions for existing software.

✓What is the JVM?

The Java Virtual Machine (JVM) is an abstract computing machine that enables a computer to run a Java program. It converts bytecode into machine code.

✓What is the JDK?

The Java Development Kit (JDK) is a software development kit used to develop Java applications. It includes the JRE, an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), and other tools needed for Java development.

✓What is the JRE?

The Java Runtime Environment (JRE) is a set of software tools for development of Java applications. It combines the Java Virtual Machine (JVM), platform core classes, and supporting libraries.

✓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. JIT is a part of JVM, JIT is responsible for compiling bytecode into native machine code at run time.

✓What is the difference between a Local variable and an Instance/Global variable?

In Java, a local variable is typically used inside a method, constructor, or a block and has only local scope. Thus, this variable can be used only within the scope of a block. The best benefit of having a local variable is that other methods in the class won't be even aware of that variable.

Instance/Global variable in Java, is a variable which is bounded to its object itself. These variables are declared within a class, but outside a method. Every object of that class will create its own copy of the variable while using it. Thus, any changes made to the variable won't reflect in another instances of that class and will be bound to that particular instance only.

✓What are the data types supported in Java?

Primitive data types: byte, short, int, long, float, double, char, boolean.

Reference/Object data types: Strings, Arrays, Classes, Interfaces.

✓What is type casting? Difference between implicit and explicit casting?

Type casting is the process of converting one data type to another.

Implicit casting (widening): Automatically done by the compiler, converting a smaller type to a larger type. Example: int to long.

Explicit casting (narrowing): Manually done by the programmer, converting a larger type to a smaller type. Example: double to int, done using a cast operator: (int) myDouble.

✓What is the purpose of the final keyword when used with variables?

When the final keyword is used with a variable, it makes the variable's value constant. This means once a final variable is initialized, its value cannot be changed. Example: final int MAX = 100;.

final: variable: Its value cannot be changed.

final: method: Cannot be overridden.

final: class: Cannot be subclassed.

✓What are the data types in Java?

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

Primitive Data Type

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:

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

boolean: stores value true or false

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

✓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.

✓What are comments in Java? Types of comments in Java?

Comments are annotations in the source code that are ignored by the compiler. They are used to make

the code more readable and to explain the code to other developers. Comments improve code readability, making it easier for others to understand, maintain, and extend the code. They are essential for collaborative development and code reviews.

There are three types of comments in Java:

Single-line comments: `// comment`

Multi-line comments: `/* comment */`

Documentation comments: `/** comment */`

✓What are the different types of errors in Java?

There are three main types of errors in Java:

Syntax errors (compile-time errors)

Runtime errors

Logical errors

Syntax errors occur when the code violates the syntax rules of the Java language, such as missing semicolons, incorrect use of keywords, or mismatched parentheses. These errors are detected at compile time.

Runtime errors occur during the execution of a program. They are usually caused by illegal operations, such as dividing by zero, accessing invalid array indices, or attempting to use null references.

Logical errors occur when the code compiles and runs without crashing, but the output is incorrect or does not meet the expected result due to flaws in the algorithm or logic used.

An Exception in Java is an event that disrupts the normal flow of the program's instructions. It is an object that is thrown at runtime when an error occurs.

✓What are the types of operators available in Java?

Arithmetic Operators (+, -, \*, /, %)

Relational Operators (==, !=, >, <, >=, <=)

Logical Operators (&&, ||, !)

Bitwise Operators (&, |, ^, ~, <<, >>, >>>)

Assignment Operators (=, +=, -=, \*=, /=, %=)

Unary Operators (+, -, ++, --)

Ternary Operator (? :)

✓How does the && operator differ from the & operator?

The && operator is a short-circuit logical AND operator. It only evaluates the right-hand side operand if the left-hand side operand is true. The & operator is a bitwise AND operator when used with integral types and a logical AND operator without short-circuiting when used with boolean types.

✓How do the bitwise shift operators <<, >>, and >>> work?

<< (left shift) shifts the bits of a number to the left, filling the rightmost bits with zeros.

>> (right shift) shifts the bits of a number to the right, preserving the sign bit (sign extension).

>>> (unsigned right shift) shifts the bits of a number to the right, filling the leftmost bits with zeros (zero extension).

✓What is the ternary operator in Java and how is it used?

The ternary operator (? :) is a shorthand for an if-else statement. It is used to evaluate a boolean expression and return one of two values based on the result. The syntax is: condition ? value1 : value2. If condition is true, value1 is returned; otherwise, value2 is returned.

Can you explain the difference between ++i and i++?

Both ++i (pre-increment) and i++ (post-increment) increment the value of i by 1. The difference is in their return value:

++i increments i and then returns the incremented value.

i++ returns the current value of i and then increments i.

✓What is operator precedence, and why is it important in Java?

Operator precedence determines the order in which operators are evaluated in an expression. Understanding operator precedence is important to predict the outcome of complex expressions correctly. For example, in the expression  $a + b * c$ , the multiplication (\*) has higher precedence than addition (+), so  $b * c$  is evaluated first.

✓What is an infinite loop in Java? Explain with an example.

An infinite loop is an instruction sequence in Java that loops endlessly when a functional exit isn't met. This type of loop can be the result of a programming error or may also be a deliberate action based on the application behavior. An infinite loop will terminate automatically once the application exits.

✓What is an Array in Programming?

An Array is a collection of similar data types stored in contiguous memory locations. An array in Java is a data structure that holds a fixed number of values of the same type. At the time of declaration of an array, you must specify the type of data with the array name.

You can access different elements present in an array using their index. For example, if you want to access an element present at the 3rd index(4th element) in an array arr, then you can write `arr[3]`.

Declaration: `int[] array;`

Initialization: `array = new int[10];` or `int[] array = new int[]{1, 2, 3, 4, 5};`

✓Mention some advantages and disadvantages of Arrays.

Advantages:

Multiple elements of Array can be sorted at the same time.

Using the index, we can access any element in  $O(1)$  time.

Disadvantages:

You need to specify how many elements you're going to store in your array ahead of time and We can not increase or decrease the size of the Array after creation.

You have to shift the other elements to fill in or close gaps, which takes worst-case  $O(n)$  time.

✓When will we get `ArrayIndexOutOfBoundsException` Exception?

`ArrayIndexOutOfBoundsException` is a runtime exception that occurs when the program tries to access the invalid index of an array such as an Index higher than the size of the array or a negative index.

✓What is the default value of Array in Java?

If we don't specify the values by ourselves, then Java assigns default values in them which are 0 for byte, short, int, and long, 0.0 for float and double, false for boolean, and null for objects respectively.

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.

✓What are the types of an array?

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.

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

✓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.

✓What is the String class in Java?

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

✓What is the difference between String, StringBuilder, and StringBuffer?

String is immutable, meaning its value cannot be changed once created. StringBuilder and StringBuffer are mutable classes used to create mutable (modifiable) string objects. StringBuilder is faster but not thread-safe, while StringBuffer is thread-safe.

✓What is Java StringPool?

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.

✓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");

✓What is Object Oriented Programming?

Object-oriented programming, or OOP, is a programming model or approach in which programs are organized around objects rather than logic and functions. In other words, OOP mainly focuses on the objects that need to be manipulated instead of logic. This approach is ideal for large and complex programs and needs to be actively updated or maintained.

✓What are the main concepts of OOPs in Java?

Object-Oriented Programming or OOPs is a programming style that is associated with concepts like:

Inheritance: Inheritance is a process where one class acquires the properties of another.

Encapsulation: Encapsulation in Java is a mechanism of wrapping up the data and code together as a single unit.

Abstraction: Abstraction is the methodology of hiding the implementation details from the user and only providing the functionality to the users.

Polymorphism: Polymorphism is the ability of a variable, function or object to take multiple forms.

✓What is the difference between public, protected, default, and private access modifiers?

public: Accessible from any other class.

protected: Accessible within the same package and subclasses.

Default (no modifier): Accessible only within the same package.

private: Accessible only within the same class.

✓ 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:

Constructors are only called when the object is created but other methods can be called multiple times during the life of an object.

Constructors do not have a return type, whereas methods have a return type, which can be void or any other type.

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

✓ What is a copy constructor in Java?

Copy constructor is a member function that is used to initialize an object using another object of the same class. Though there is no need for copy constructor in Java since all objects are passed by reference.

Moreover, Java does not even support automatic pass-by-value.

✓ What is inheritance in Java?

Inheritance is a mechanism wherein a new class is derived from an existing class. The new class inherits the properties and behaviors of the existing class.

- ✓ 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.
- ✓ 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.  
Note: Java doesn't support Multiple Inheritance.
- ✓ Can the constructor be inherited?  
No, we can't inherit a constructor.  
How can we restrict inheritance for a class?  
We can restrict inheritance for a class by the following steps:  
By using the final keyword  
If we make all methods final, then we cannot override that  
By using private constructors  
By using the Javadoc comment (" // ")
- ✓ What is method overloading?  
In Java, Method Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters, or a mixture of both. Method overloading in Java is also known as Compile-time Polymorphism, Static Polymorphism, or Early binding. In Method overloading compared to the parent argument, the child argument will get the highest priority.
- ✓ What is method overriding?  
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.
- ✓ 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.
- ✓ 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.
- ✓ What is polymorphism in Java?  
Polymorphism in Java allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation.
- ✓ What is encapsulation?  
Encapsulation is the technique of making the fields in a class private and providing access to them via

public methods. It helps to protect the data from unauthorized access and modification. 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.

✓ What are the advantages of Encapsulation in Java?

The advantages of Encapsulation in Java are mentioned below:

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.

Increased Flexibility: We can make the variables of the class read-only or write-only depending on our requirements.

Reusability: Encapsulation also improves the re-usability and is easy to change with new requirements.

Testing code is easy: Code is made easy to test for unit testing.

✓ 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.

✓ What is an abstract class?

An abstract class is a class that cannot be instantiated on its own and is designed to be subclassed. It may contain abstract methods, which are methods declared without an implementation.

✓ 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.

✓ What is an interface?

An interface in Java is a reference type, similar to a class, that can contain only constants, method signatures, default methods, static methods, and nested types. Interfaces cannot contain instance fields or constructors.

✓ What is the difference between an abstract class and an interface?

An abstract class can have instance methods that implement a default behavior. Interfaces can only declare methods, and default methods, but cannot provide implementation. A class can implement multiple interfaces but can extend only one class.

✓ What is a package in Java?

A package is a namespace that organizes a set of related classes and interfaces. Packages are used to avoid name conflicts and to control access to classes, interfaces, and other components.

There are two types of packages in Java

User-defined packages

Build In packages

✓ 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.

✓ What is exception handling in Java?

Exception handling in Java is a mechanism to handle runtime errors, allowing the program to continue its normal flow. It is done using five keywords: try, catch, finally, throw, and throws.

✓ What is the difference between checked and unchecked exceptions?

### 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.

#### ✓ 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.

#### ✓ What are the benefits of exception handling?

Exception handling helps to maintain the normal flow of the application, separates error-handling code from regular code, provides a method for reporting error conditions, and allows handling multiple exceptions.

`NullPointerException` is thrown when the JVM attempts to use an object reference that has not been initialized or is null. Common causes include calling methods on null objects, accessing fields of null objects, or attempting to use null as an array.

`ArrayIndexOutOfBoundsException` occurs when an application tries to access an array with an index that is outside the valid range (i.e., less than 0 or greater than or equal to the array's length).

`ArithmeticException` is thrown when an exceptional arithmetic condition has occurred. A common cause is division by zero.

`ClassCastException` is thrown when an application tries to cast an object to a subclass of which it is not an instance. For example, casting an `Integer` to a `String`.

`IllegalArgumentException` is thrown to indicate that a method has been passed an illegal or inappropriate argument. For example, passing a negative value where only positive values are expected.

`IllegalStateException` is thrown when a method is invoked at an illegal or inappropriate time. For example, calling `next()` on an iterator that has no more elements.

`IndexOutOfBoundsException` is a superclass of exceptions that are thrown when an index is out of range for arrays, strings, or collections. It includes `ArrayIndexOutOfBoundsException` and `StringIndexOutOfBoundsException`.

`NumberFormatException` is thrown when an attempt is made to convert a string to a numeric type, but the string does not have an appropriate format. For example, trying to parse "abc" into an integer.

`UnsupportedOperationException` is thrown to indicate that the requested operation is not supported by the object. For example, attempting to modify an unmodifiable collection.

`FileNotFoundException` is thrown when an attempt to open a file denoted by a specified pathname has failed. It indicates that the file does not exist.

`EOFException` (End Of File Exception) is thrown when an input operation has reached the end of a file or stream unexpectedly.

`IOException` is a general exception class that signals an I/O error has occurred. It is the superclass of all the exceptions that can occur during I/O operations.

`SQLException` is thrown when there is a database access error or other errors related to the database

operations. It is part of the java.sql package.

NoSuchElementException is thrown when one tries to access an element that is not present. For example, calling next() on an empty iterator.

ConcurrentModificationException is thrown when a collection is structurally modified while an iteration is in progress. For example, modifying a List while iterating over it.

OutOfMemoryError is a Error thrown when the JVM cannot allocate more memory. It can be due to memory leaks or insufficient heap space.

NoClassDefFoundError is thrown if the JVM cannot find the definition of a class that was present during the compilation but is not available at runtime.

ClassNotFoundException is thrown when an application tries to load a class through its name but cannot find it. It typically occurs with dynamic class loading using reflection.

- ✓ What purpose do the keywords final, finally, and finalize fulfill?

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

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.

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

- ✓ What is garbage collection in Java?

Garbage collection in Java is the process of reclaiming the runtime unused memory automatically. It is a form of automatic memory management.

- ✓ 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.

- ✓ What is the difference between abstract class and interface in Java 8?

Abstract Class: Can have abstract and non-abstract methods.

Interface: Can have default and static methods in addition to abstract methods.

- ✓ What is the difference between an abstract class and an interface?

An abstract class is a class that cannot be instantiated and may contain both abstract and non-abstract methods, while an interface is a collection of abstract methods that must be implemented by any class that implements it.

- ✓ What is the Java Virtual Machine (JVM)?

The JVM is an abstract computing machine that provides the runtime environment for Java programs. It interprets compiled Java code and executes it on the computer hardware.

- ✓ What is a package in Java?

A package in Java is a mechanism for organizing classes and interfaces into namespaces. It helps to prevent naming conflicts and makes it easier to locate and use classes in a large project.

- ✓ What is the difference between public, private, and protected access modifiers in Java?

Public, private, and protected are access modifiers in Java that determine the accessibility of class members. Public members can be accessed from anywhere, private members can only be accessed within the class, and protected members can be accessed within the class and its subclasses.

- ✓ What is a static method in Java? Java Interview Questions and Answers

A static method in Java is a method that belongs to the class rather than to any specific instance of the

class. It can be called using the class name, rather than an instance of the class.

✓ What is a final keyword in Java?

The final keyword in Java can be used to indicate that a class cannot be extended, a method cannot be overridden, or a variable cannot be reassigned. It is often used to define constants.

✓ What is a thread in Java?

A thread in Java is a lightweight process that allows a program to execute multiple tasks concurrently. Each thread has its own call stack, but shares the same memory and resources as other threads in the program.

✓ What is a try-catch block in Java?

A try-catch block in Java is used for exception handling. The code that may throw an exception is placed in the try block, and any exceptions that are thrown are caught and handled in the catch block.

✓ What is the difference between checked and unchecked exceptions in Java?

Checked exceptions are checked at compile-time and must be handled by the programmer, while unchecked exceptions are not checked at compile-time and can be handled at runtime if necessary.

✓ What is the difference between a private and a protected access modifier in Java?

A private access modifier in Java restricts access to the member to only within the same class, (<https://fullstackadda.com/>) while a protected access modifier allows access to the member within the same class and its subclasses.

✓ What is the difference between a static and a non-static method in Java?

A static method in Java is a method that belongs to the class and can be called without an instance of the class, while a non-static method is a method that belongs to an instance of the class.

✓ What is a package in Java? What is the purpose of a package in Java?

A package in Java is a namespace that organizes a set of related classes and interfaces.

A package in Java is a way of organizing related classes and interfaces into a single unit, making it easier to manage and maintain large code bases. Packages also help to avoid naming conflicts between classes and provide a level of access control to the classes within them.

✓ What is the difference between a private and a protected method in Java?

A private method can only be accessed within the same class, while a protected method can be accessed within the same class and any subclasses. Java Interview Questions and Answers

✓ How is memory allocation managed in Java?

Memory allocation in Java is managed by the JVM, which allocates memory on the heap for objects and on the stack for local variables and method calls. Java Interview Questions and Answers

✓ What is the purpose of the “this” keyword in Java?

The “this” keyword in Java refers to the current object and is used to refer to instance variables and methods within the class.

\*\*\*\*\*