I'm not a robot

Here are Java interview questions and answers for fresher as well as experienced candidates to get their dream job. Java Programming Interview Questions and Answers for freshers 1) What is the difference between an Inner Class and a Sub-Class? An Inner class is a class which is nested within another class. An Inner class has access rights for the class which is nesting it and it can access all variables and methods defined in the outer class. A sub-class is a class which inherits from another class called super class. Sub-class can access all public and protected methods and fields of its super class. Free PDF Download: Java Interview Interview Questions & Answers >> 2) What are the various access specifiers for Java classes? In Java, access specifiers are the keywords used before a class name which defines the access scope. The types of access specifiers for classes are: 1) Public: Class,Method,Field is accessible from anywhere. 2) Protected: Method,Field can be accessed from the same class to which they belong or from the sub-classes, and from the class of same package, but not from outside. 3) Default: Method,Field,class can be accessed only from the same package and not from outside of it's native package. 4) Private: Method,Field can be accessed from the same class to which they belong. 3) What's the purpose of Static methods and static variables? When there is a requirement to share a method or a variable between multiple objects of a class instead of creating separate copies for each object, we use static keyword to make a method or variable shared for all objects. 4) What is data encapsulation and what's its significance? Encapsulation is a concept in Object Oriented Programming for combining properties and methods in a single unit. Encapsulation helps programmers to follow a modular approach for software development as each object has its own set of methods and variables and serves its functions independent of other objects. Encapsulation also serves data hiding purpose. 5) What is a singleton class? Give a practical example of its usage. A singleton class in java can have only one instance and hence all its methods and variables belong to just one instance. Singleton class concept is useful for the situations when there is a need to limit the number of objects for a class. The best example of singleton usage scenario is when there is a limit of having only one connection to a database due to some driver limitations or because of any licensing issues. 6) What are Loops in Java? What are three types of loops? Looping is used in programming to execute a statement or a block of statement repeatedly. There are three types of loops in Java: 1) For Loops For loops are used in java to execute statements repeatedly for a given number of times. For loops are used when number of times to execute the statements is known to programmer. 2) While Loops While loop is used when certain statements need to be executed repeatedly until a condition is fulfilled. In while loops, condition is checked first before execution of statements. 3) Do While Loops Do While Loop is same as While loop with only difference that condition is checked after execution of block of statements. Hence in case of do while loop, statements are executed at least once. 7) What is an infinite Loop? How infinite loop is declared? An infinite loop runs without any condition and runs infinitely. An infinite loop can be broken by defining any breaking logic in the body of the statement blocks. Infinite loop is declared as follows: for (;;) { // Statements to execute // Add any loop breaking logic } 8) What is the difference between continue and break statement? break and continue are two important keywords used in Loops. When a break keyword is used in a loop, loop is broken instantly while when continue keyword is used, current iteration is broken and loop continues with next iteration. In below example, Loop is broken when counter reaches 4. for (counter = 0; counter & lt; 10; counter++) system.out.println(counter); if (counter == 4) { break; } } In the below example when counter reaches 4, loop jumps to next iteration and any statements after the continue keyword are skipped for current iteration. for (counter = 0; counter < 10; counter++) system.out.println(counter); if (counter == 4) { continue; } system.out.println("This will not get printed when counter is 4"); } 9) What is the difference between double and float variables in Java? In java, float takes 4 bytes in memory while Double takes 8 bytes in memory. Float is single precision floating point decimal number while Double is double precision decimal number. 10) What is Final Keyword in Java? Give an example. In java, a constant is declared using the keyword Final. Value can be assigned only once and after assignment, value of a constant can't be changed. In below example, a constant with the name const_val is declared and assigned avalue: Private Final int const_val=100 When a method is declared as final, it can NOT be overridden by the subclasses. This method are faster than any other method, because they are resolved at complied time. When a class is declares as final, it cannot be subclassed. Example String, Integer and other wrapper classes. 11) What is ternary operator? Give an example. Ternary operator , also called conditional operator is used to decide which value to assign to a variable based on a Boolean value evaluation. It's denoted as ? In the below example, if rank is 1, status is assigned a value of "Done" else "Pending". public class conditionTest { public static void main(String args[]) { String status; int rank = 3; status = (rank == 1) ? "Done" : "Pending"; System.out.println(status); } } 12) How can you generate random numbers in Java? Using Math.random() you can generate random numbers in the range greater than or equal to 0.1 and less than 1.0 Using Random class in package java.util 13) What is default switch case? Give example. In a switch statement, default case is executed when no other switch condition matches. Default case is an optional case .It can be declared only once all other switch cases have been coded. In the below example, when score is not 1 or 2, default case is used. public class SwitchExample { public static void main(String args[]) { int score = 4; // Declare the score variable inside the main method switch (score) { case 1: System.out.println("Score is 1"); break; case 2: System.out.println("Score is 2"); break; default: System.out.println("Default Case"); } } } 14) What's the base class in Java from which all classes are derived? java.lang.object 15) Can main() method in Java can return any data? In java, main() method can't return any data and hence, it's always declared with a void return type. 16) What are Java Packages? What's the significance of packages? In Java, package is a collection of classes and interfaces which are bundled together as they are related to each other. Use of packages helps developers to modularize the code and group the code for proper re-use. Once code has been packaged in Packages, it can be imported in other classes and used. 17) Can we declare a class as Abstract without having any abstract method? Yes we can create an abstract class by using abstract keyword before class name even if it doesn't have any abstract method. However, if a class has even one abstract method, it must be declared as abstract otherwise it will give an error. 18) What's the difference between an Abstract Class and Interface in Java? The primary difference between an abstract class and interface is that an interface can only possess declaration of public static methods with no concrete implementation while an abstract class can have members with any access specifiers (public, private etc) with or without concrete implementation. Another key difference in the use of abstract classes and interfaces is that a class which implements an interface must implement all the methods of the interface while a class which inherits from an abstract class doesn't require implementation of all the methods of its super class. A class can implement multiple interfaces but it can extend only one abstract class. 19) What are the performance implications of Interfaces over abstract classes? Interfaces are slower in performance as compared to abstract classes as extra indirections are required for interfaces. Another key factor for developers to take into consideration is that any class can extend only one abstract class while a class can implement many interfaces. Use of interfaces also puts an extra burden on the developers as any time an interface is implemented in a class; developer is forced to implement each and every method of interface. 20) Does Importing a package imports its sub-packages as well in Java? In java, when a package is imported, its sub-packages aren't imported and developer needs to import them separately if required. For example, if a developer imports a package university.*, all classes in the package named university are loaded but no classes from the sub-package are loaded. To load the classes from its sub-package (say department), developer has to import it explicitly as follows: Import university.department.* 21) Can we declare the main method of our class as private? In java, main method must be public static in order to run any application correctly. If main method is declared as private, developer won't get any compilation error however, it will not get executed and will give a runtime error. 22) How can we pass argument to a function by reference instead of pass by value? In java, we can pass argument to a function only by value and not by reference. 23) How an object is serialized in java? In java, to convert an object into byte stream by serialization, an interface with the name Serializable is implemented by the class. All objects of a class implementing serializable interface get serialized and their state is saved in byte stream. 24) When we should use serialization? Serialization is used when data needs to be transmitted over the network. Using serialization, object's state is saved and converted into byte stream .The byte stream is transferred over the network and the object is re-created at destination. 25) Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling? Try block needs to be followed by either Catch block or Finally block or both. Any exception thrown from try block needs to be either caught in the catch block or else any specific tasks to be performed before code abortion are put in the Finally block. Java Interview Questions and Answers for Experienced 26) Is there any way to skip Finally block of exception even if some exception occurs in the exception block? If an exception is raised in Try block, control passes to catch block if it exists otherwise to finally block. Finally block is always executed when an exception occurs and the only way to avoid execution of any statements in Finally block is by aborting the code forcibly by writing following line of code at the end of try block: System.exit(0); 27) When the constructor of a class is invoked? The constructor of a class is invoked every time an object is created with new keyword. For example, in the following class two objects are created using new keyword and hence, constructor is invoked two times. public class const_example { const_example() { system.out.println("Inside constructor"); } public static void main(String args[]) { const_example c1 = new const_example(); const_example c2 = new const_example(); } } 28) Can a class have multiple constructors? Yes, a class can have multiple constructors with different parameters. Which constructor gets used for object creation depends on the arguments passed while creating the objects. 29) Can we override static methods of a class? We cannot override static methods. Static methods belong to a class and not to individual objects and are resolved at the time of compilation (not at runtime).Even if we try to override static method, we will not get an complitaion error,nor the impact of overriding when running the code. 30) In the below example, what will be the output? public class superclass { public void displayResult() { system.out.println("Printing from superclass"); } } public class subclass extends superclass { public void displayResult() { system.out.println("Displaying from subClass"); super.displayResult(); } public static void main(String args[]) { subclass obj = new subclass(); obj.displayResult(); } } Ans: Output will be: Displaying from subClass Printing from superclass 31) Is String a data type in java? String is not a primitive data type in java. When a string is created in java, it's actually an object of Java.Lang.String class that gets created. After creation of this string object, all built-in methods of String class can be used on the string object. 32) In the below example, how many String Objects are created? String s1="I am Java Expert"; String s2="I am C Expert"; String s3="I am Java Expert"; In the above example, two objects of Java.Lang.String class are created. s1 and s3 are references to same object. 33) Why Strings in Java are called as Immutable? In java, string objects are called immutable as once value has been assigned to a string, it can't be changed and if changed, a new object is created. In below example, reference str refers to a string object having value "Value one". String str="Value One"; When a new value is assigned to it, a new String object gets created and the reference is moved to the new object. str="New Value"; 34) What's the difference between an array and Vector? An array groups data of same primitive type and is static in nature while vectors are dynamic in nature and can hold data of different data types. 35) What is multi-threading? Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program. 36) Why Runnable Interface is used in Java? Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading. 37) What are the two ways of implementing multi-threading in Java? Multi threaded applications can be developed in Java by using any of the following two methodologies: 1) By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented. 2) By writing a class that extend Java.Lang.Thread class. 38) When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer? Since StringBuffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it's always a good choice to use StringBuffer when data is being changed too much. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead. 39) What's the purpose of using Break in each case of Switch Statement? Break is used after each case (except the last one) in a switch so that code breaks after the valid case and doesn't flow in the proceeding cases too. If break isn't used after each case, all cases after the valid case also get executed resulting in wrong results. 40) How garbage collection is done in Java? In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection java calls either System.gc() method or Runtime.gc() method. 41) How we can execute any code even before main method? If we want to execute any statements before even creation of objects at load time of class, we can use a static block of code in the class. Any statements inside this static block of code will get executed once at the time of loading the class even before creation of objects in the main method. 42) Can a class be a super class and a sub-class at the same time? Give example. If there is a hierarchy of inheritance used, a class can be a super class for another class and a sub-class for another one at the same time. In the example below, continent class is sub-class of world class and it's super class of country class. public class world { .... } public class continenet extends world { .... } public class country extends continent { .... } 43) How objects of a class are created if no constructor is defined in the class? Even if no explicit constructor is defined in a java class, objects get created successfully as a default constructor is implicitly used for object creation. This constructor has no parameters. 44) In multi-threading how can we ensure that a resource isn't used by multiple threads simultaneously? In multi-threading, access to the resources which are shared among multiple threads can be controlled by using the concept of synchronization. Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it has become free from the other one using it. 45) Can we call the constructor of a class more than once for an object? Constructor is called automatically when we create an object using new keyword. It's called only once for an object at the time of object creation and hence, we can't invoke the constructor again for an object after its creation. 46) There are two classes named classA and classB. Both classes are in the same package. Can a private member of classA can be accessed by an object of classB? Private members of a class aren't accessible outside the scope of that class and any other class even in the same package can't access them. 47) Can we have two methods in a class with the same name? We can define two methods in a class with the same name but with different number/type of parameters. Which method is to get invoked will depend upon the parameters passed. For example in the class below we have two print methods with same name but different parameters. Depending upon the parameters, appropriate one will be called: public class methodExample { public void print() { system.out.println("Print method without parameters."); } public void print(String name) { system.out.println("Print method with parameter"); } public static void main(String args[]) { methodExample obj1 = new methodExample(); obj1.print(); obj1.print("xx"); } } 48) How can we make copy of a java object? We can use the concept of cloning to create copy of an object. Using clone, we create copies with the actual state of an object. Clone() is a method of Cloneable interface and hence, Cloneable interface needs to be implemented for making object copies. 49) What's the benefit of using inheritance? Key benefit of using inheritance is reusability of code as inheritance enables sub-classes to reuse the code of its super class. Polymorphism (Extensibility) is another great benefit which allow new functionality to be introduced without effecting existing derived classes. 50) What's the default access specifier for variables and methods of a class? Default access specifier for variables and method is package protected i.e variables and class is available to any other class but in the same package, not outside the package. 51) Give an example of use of Pointers in Java class. There are no pointers in Java. So we can't use concept of pointers in Java. 52) How can we restrict inheritance for a class so that no class can be inherited from it? If we want a class not to be extended further by any class, we can use the keyword Final with the class name. In the following example, Stone class is Final and can't be extend public Final Class Stone { // Class methods and Variables } 53) What's the access scope of Protected Access specifier? When a method or a variable is declared with Protected access specifier, it becomes accessible in the same class, any other class of the same package as well as a sub-class. Modifier Class Package Subclass World public Y Y Y Y protected Y Y Y N no modifier Y Y N N private Y N N N 54) What's difference between Stack and Queue? Stack and Queue both are used as placeholder for a collection of data. The primary difference between a stack and a queue is that stack is based on Last in First out (LIFO) principle while a queue is based on FIFO (First In First Out) principle. 55) In java, how we can disallow serialization of variables? If we want certain variables of a class not to be serialized, we can use the keyword transient while declaring them. For example, the variable trans_var below is a transient variable and can't be serialized: public class transientExample { private transient trans_var; // rest of the code } 56) How can we use primitive data types as objects? Primitive data types like int can be handled as objects by the use of their respective wrapper classes. For example, Integer is a wrapper class for primitive data type int. We can apply different methods to a wrapper class, just like any other object. 57) Which types of exceptions are caught at compile time? Checked exceptions can be caught at the time of program compilation. Checked exceptions must be handled by using try catch block in the code in order to successfully compile the code. 58) Describe different states of a thread. A thread in Java can be in either of the following states: Ready: When a thread is created, it's in Ready state. Running: A thread currently being executed is in running state. Waiting: A thread waiting for another thread to free certain resources is in waiting state. Dead: A thread which has gone dead after execution is in dead state. 59) Can we use a default constructor of a class even if an explicit constructor is defined? Java provides a default no argument constructor if no explicit constructor is defined in a Java class. But if an explicit constructor has been defined, default constructor can't be invoked and developer can use only those constructors which are defined in the class. 60) Can we override a method by using same method name and arguments but different return types? The basic condition of method overriding is that method name, arguments as well as return type must be exactly same as is that of the method being overridden. Hence using a different return type doesn't override a method. 61) What will be the output of following piece of code? public class operatorExample { public static void main(String args[]) { int x = 4; system.out.println(x++); } } In this case postfix ++ operator is used which first returns the value and then increments. Hence it's output will be 4. 61) A person says that he compiled a java class successfully without even having a main method in it? Is it possible? main method is an entry point of Java class and is required for execution of the program however; a class gets compiled successfully even if it doesn't have a main method. It can't be run though. 62) Can we call a non-static method from inside a static method? Non-Static methods are owned by objects of a class and have object level scope and in order to call the non-Static methods from a static block (like from a static main method), an object of the class needs to be created first. Then using object reference, these methods can be invoked. 63) What are the two environment variables that must be set in order to run any Java programs? Java programs can be executed in a machine only once following two environment variables have been properly set: PATH variable CLASSPATH variable 64) Can variables be used in Java without initialization? In Java, if a variable is used in a code without prior initialization by a valid value, program doesn't compile and gives an error as no default value is assigned to variables in Java. 65) Can a class in Java be inherited from more than one class? In Java, a class can be derived from only one class and not from multiple classes. Multiple inheritances is not supported by Java. 66) Can a constructor have different name than a Class name in Java? Constructor in Java must have same name as the class name and if the name is different, it doesn't act as a constructor and compiler thinks of it as a normal method. 67) What will be the output of Round(3.7) and Ceil(3.7)? Round(3.7) returns 4 and Ceil(3.7) returns 4. 68) Can we use goto in Java to go to a particular line? In Java, there is not goto keyword and java doesn't support this feature of going to a particular labeled line. 69) Can a dead thread be started again? In java, a thread which is in dead state can't be started again. There is no way to restart a dead thread. 70) Is the following class declaration correct? public abstract final class testClass { // Class methods and variables } Ans: The above class declaration is incorrect as an abstract class can't be declared as Final. 71) Is JDK required on each machine to run a Java program? JDK is development Kit of Java and is required for development only and to run a Java program on a machine, JDK isn't required. Only JRE is required. 72) What's the difference between comparison done by equals method and == operator? In Java, equals() method is used to compare the contents of two string objects and returns true if the two have same value while == operator compares the references of two string objects. In the following example, equals() returns true as the two string objects have same values. However == operator returns false as both string objects are referencing to different objects: public class equalsTest { public static void main(String args[]) { String str1 = new String("Hello World"); String str2 = new String("Hello World"); if (str1.equals(str2)) { // this condition is true System.out.println("str1 and str2 are equal in terms of values"); } if (str1 == str2) { //This condition is true System.out.println("Both strings are referencing same object"); } else { // This condition is NOT true System.out.println("Both strings are referencing different objects"); } } } 73) Is it possible to define a method in Java class but provide it's implementation in the code of another language like C? Yes, we can do this by use of native methods. In case of native method based development, we define public static methods in our Java class without its implementation and then implementation is done in another language like C separately. 74) How are destructors defined in Java? In Java, there are no destructors defined in the class as there is no need to do so. Java has its own garbage collection mechanism which does the job automatically by destroying the objects when no longer referenced. Java Interview Questions and Answers for 5+ Years Experience 75) Can a variable be local and static at the same time? No a variable can't be static as well as local at the same time. Defining a local variable as static gives compilation error. 76) Can we have static methods in an Interface? Static methods can't be overridden in any class while any methods in an interface are by default abstract and are supposed to be implemented in the classes being implementing the interface. So it makes no sense to have static methods in an interface in Java. 77) In a class implementing an interface, can we change the value of any variable defined in the interface? No, we can't change the value of any variable of an interface in the implementing class as all variables defined in the interface are by default public, static and Final and final variables are like constants which can't be changed later. 78) Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory? Even though automatic garbage collection is provided by Java, it doesn't ensure that a Java program will not go out of memory as there is a possibility that creation of Java objects is being done at a faster pace compared to garbage collection resulting in filling of all the available memory resources. So, garbage collection helps in reducing the chances of a program going out of memory but it doesn't ensure that. 79) Can we have any other return type than void for main method? No, Java class main method can have only void return type for the program to get successfully executed. Nonetheless , if you absolutely must return a value to at the completion of main method , you can use System.exit(int status) 80) I want to re-reach and use an object once it has been garbage collected. How it's possible? Once an object has been destroyed by garbage collector, it no longer exists on the heap and it can't be accessed again. There is no way to reference it again. 81) In Java thread programming, which method is a must implementation for all threads? Run() is a method of Runnable interface that must be implemented by all threads. 82) I want to control database connections in my program and want that only one thread should be able to make database connection at a time. How can I implement this logic? Ans: This can be implemented by use of the concept of synchronization. Database related code can be placed in a method which hs synchronized keyword so that only one thread can access it at a time. 83) How can an exception be thrown manually by a programmer? In order to throw an exception in a block of code manually, throw keyword is used. Then this exception is caught and handled in the catch block. public void topMethod() { try { excMethod(); } catch (ManualException e) {} } public void excMethod { String name = null; if (name == null) { throw (new ManualException("Exception thrown manually "); } } 84) I want my class to be developed in such a way that no other class (even derived class) can create its objects. How can I do so? If we declare the constructor of a class as private, it will not be accessible by any other class and hence, no other class will be able to instantiate it and formation of its object will be limited to itself only. 85) How objects are stored in Java? In java, each object when created gets a memory space from a heap. When an object is destroyed by a garbage collector, the space allocated to it from the heap is re-allocated to the heap and becomes available for any new objects. 86) How can we find the actual size of an object on the heap? In java, there is no way to find out the exact size of an object on the heap. 87) Which of the following classes will have more memory allocated? Class A: Three methods, four variables, no object Class B: Five methods, three variables, no object Memory isn't allocated before creation of objects. Since for both classes, there are no objects created so no memory is allocated on heap for any class. 88) What happens if an exception is not handled in a program? If an exception is not handled in a program using try catch blocks, program gets aborted and no statement executes after the statement which caused exception throwing. 89) I have multiple constructors defined in a class. Is it possible to call a constructor from another constructor's body? If a class has multiple constructors, it's possible to call one constructor from the body of another one using this(). 90) What's meant by anonymous class? An anonymous class is a class defined without any name in a single line of code using new keyword. For example, in below code we have defined an anonymous class in one line of code: public java.util.Enumeration testMethod() { return new java.util.Enumeration() { @Override public boolean hasMoreElements() { // TODO Auto-generated method stub return false; } @Override public Object nextElement() { // TODO Auto-generated method stub return null; } } 91) Is there a way to increase the size of an array after its declaration? Arrays are static and once we have specified its size, we can't change it. If we want to use such collections where we may require a change of size (no of items), we should prefer vector over array. 92) If an application has multiple classes in it, is it okay to have a main method in more than one class? If there is main method in more than one classes in a java application, it won't cause any issue as entry point for any application will be a specific class and code will start from the main method of that particular class only. 93) I want to persist data of objects for later use. What's the best approach to do so? The best way to persist data for future use is to use the concept of serialization. 94) What is a Local class in Java? In Java, if we define a new class inside a particular block, it's called a local class. Such a class has local scope and isn't usable outside the block where its defined. 95) String and StringBuffer both represent String objects. Can we compare String and StringBuffer in Java? Although String and StringBuffer both represent String objects, we can't compare them with each other and if we try to compare them, we get an error. 96) Which API is provided by Java for operations on set of objects? Java provides a Collection API which provides many useful methods which can be applied on a set of objects. Some of the important classes provided by Collection API include ArrayList, HashMap, TreeSet and TreeMap. 97) Can we cast any other type to Boolean Type with type casting? No, we can neither cast any other primitive type to Boolean data type nor can cast Boolean data type to any other primitive data type. 98) Can we use different return types for methods when overridden? The basic requirement of method overriding in Java is that the overridden method should have same name, and parameters.But a method can be overridden with a different return type as long as the new return type extends the original. For example , method is returning a reference type. Class B extends A { A method(int x) { //original method } B method(int x) { //overridden method } } 99) What's the base class of all exception classes? In Java, Java.lang.Throwable is the super class of all exception classes and all exception classes are derived from this base class. 100) What's the order of call of constructors in inheritance? In case of inheritance, when a new object of a derived class is created, first the constructor of the super class is invoked and then the constructor of the derived class is invoked. Prep Up For your Job Interview!!! Go through Java Tutorial to be better prepared. This detailed Java Mock Test Quiz will help you to clear the doubts about Java interview questions and will also help you to crack the interview. These Java and Java 8 interview questions will also help in your viva(orals) Skip to content Accessibility Policy Oracle Java is the #1 programming language and development platform. It reduces costs, shortens development timeframes, drives innovation, and improves application services. With millions of developers running more than 60 billion Java Virtual Machines worldwide, Java continues to be the development platform of choice for enterprises and developers. Assess the health of your Java environment Java 24 is now available The next Java release improves the performance, stability, and security of Java application development. Oracle GraalVM free on OCI Build native executables that help applications start up fast, reduce memory usage, and save hosting costs. Java Management Service JMS Advance features help administrators gain additional insights into Java workloads. Analyze usage, vulnerabilities, and impact from Cryptographic Roadmap updates. Transforming Development for Next-Generation Software Supply Chains Get executive insights on leading trends and challenges impacting development organizations today, along with technology choices such as Java, that can help address them efficiently based on a recent VDC Research study. Get the VDC Research report Top IT security and compliance pain points in application development Security remains the most important priority for IT executives, according to a recent report from 451 Research. Read the brief to find out the top security concerns for developers and how you can achieve your IT security and compliance goals with Java. Get the 451 Research brief Aberdeen Knowledge Brief Learn why leaders in application development use Java to more efficiently build world-class applications the highest quality and the strongest security. Case study: Oracle Java EPP for Oracle Fusion Find out why Oracle Java Enterprise Performance Pack helps Oracle Fusion Applications improve application response times by 40% and decrease CPU utilization by 25%. Read the case study (PDF) Take advantage of the high-performance JDK with advanced optimizations that improve Java application performance and microservices deployment—on-premises and in the cloud. Together, the two technologies add value for cloud native deployments with native image and multilanguage support. Features Advanced optimizing just-in-time compiler Ahead-of-time compiler (compile Java native executables) Seamless interoperability for polyglot applications Built on enterprise-class Oracle Java SE 24/7 Oracle Premiere Support (My Oracle Support) Oracle Cloud Infrastructure (OCI) enhances the versatility, power, and stability of Java. As the steward and leading contributor to the Java platform, Oracle continues to drive the evolution of Java in response to the demands of enterprises and to provide unparalleled expertise to support developers. Explore Java on Oracle Cloud Infrastructure Enhanced Java performance. Boost peak performance with high-performance JDK. Build and deploy native executables that start almost instantly and use fewer memory resources. Simplified Java Management Service gives you insights into all your Java deployments, on OCI and on-premises. Expert Java support from the stewards of Java to optimize Java apps whenever it makes sense. Exclusive access to patches and updates, even beyond the end of public updates. Benefits of Java SE on OCI at no additional cost. Java Card enables secure elements, such as smart cards and other tamper-resistant security chips, to host applications based on Java technology. Store and update multiple applications on a single, resource-constrained device. Features Interoperable Secure execution environment Multiapplication, multitenant Extensible and updatable "One of our largest customers was facing performance and memory issues after upgrading their system. There was huge pressure from the customer at all levels to resolve the issue as soon as possible. Once the Java Sustaining Engineering team got involved, collaborating with Oracle Support, they were able to pinpoint what was causing the issue and what changes to make in order to resolve the issue. This solution was provided very quickly, and we received kudos from the end user's executives as to the speed and efficacy of the solution. We greatly appreciate the efforts of the Java Sustaining Engineering Team and Oracle Customer Support." Takashi Hashizume, Senior Manager, AI Platform Division, NEC Corporation MIKS Limited Computer Software Company "Oracle Java SE Subscription's multilingual support team is very experienced and readily available to provide instant and parallel support helping our developers to build their projects more timely and easily with no hassles." —Mohammad Iqbal Khan, Project Manager Rothbadi & Co. IT Services "Instead of wasting time and money, we have been able to reduce overall costs by managing our Java estate with Oracle Java SE Subscription. This gives us a huge cost saving opportunity that significantly reduces our IT OPEX bills." —Fortune Nwaiwu, Business Analyst "For our professional customers, where reliability is a top priority, the small fee of Oracle Jave SE subscription is much more valuable than the many free platforms available for the returned value. We use it for many deployments, mostly for high-value solutions, where every small detail matters." —Balázs Kiss, Software Developer Corte Suprema de Justicia "The technical support Oracle provides is highly efficient and of very good quality. Their staff is trained and has the necessary experience to solve or guide in the resolution of problems raised." —Moris Mendez, Ing. de Sistemas Informaticos March 18, 2025 Sharat Chander | Senior Director, Java Product Management and Developer Relations Learn about the Java 24 release and ongoing Java innovations that address modern application development. Read the complete post | Subscribe to the blog Manage Java SE installations, updates, and upgrades across your enterprise more cost effectively. Discover the advantages of a Java license and support from the Java experts who wrote the code. Talk to a Java team member about the advantages of an Oracle Java SE Subscription. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. JVM (Java Virtual Machine) is an abstract machine, In other words, it is a program/software which takes Java bytecode and converts the byte code (line by line) into machine-understandable code. JVM acts as a run-time engine to run Java applications. JVM is the one that calls the main method present in Java code. JVM is a part of the JRE(Java Runtime Environment). Operations Performed by JVMLoading of code: The JVM loads class files that contain Java bytecode into memory. This means it finds the necessary files and gets them ready to run.Verification of code: The JVM checks the bytecode it has loaded to make sure it's correct and safe.Executing the code: The JVM runs the verified bytecode by turning it into machine code that the computer can understand.Providing a Runtime Environment: The JVM provides a runtime environment that helps manage system resources, memory, and garbage collection.ClassLoader SubsystemThe ClassLoader is a subsystem of the JVM responsible for loading class files. It primarily performs three activities:Loading: Locates and loads the class files into memory.Linking: Combines the class files and resolves dependencies.Initialization: Initializes static variables and executes static blocks.Types of Memory Areas Allocated By JVMThe JVM allocates memory into 5 distinct areas to carry out its operations. These areas are:Class(Method) AreaHeapStackProgram Counter Register (PC Register)Native Method Stack1. Class (Method) AreaThe Class (Method) Area is a memory block in the JVM that stores important information about classes. It includes:Class Code: The bytecode that defines the class itself.Variable Code: This includes static variables and runtime constants associated with the class.Method Code: The bytecode for all the methods defined in the class, including constructors.Note: Here method means the function which is written inside the classThe class area stores class-level data of every class such as the runtime constant pool, field and method data and the code for methods, enabling the JVM to execute Java programs effectively.2. HeapThe Heap Area is the memory block in the JVM where all objects are created and stored. It is used to allocate memory for:Objects: Instances of classes created during the execution of a program.Class Interfaces: Memory for class interfaces is also allocated here.Arrays: Since arrays are considered objects in Java, their memory is allocated in the heap as well.Note: Static Methods and Variables were previous stored in Class Area (Till Java 8). But, in current versions of Java static variables and methods are stored in Heap Memory.The Heap Area is crucial for dynamic memory allocation, allowing the JVM to allocate memory for objects at runtime.3. Stack In Java, each thread has its own stack called the Run-Time Stack, created when the thread starts. The JVM only performs two operations directly on Java stacks. It pushes and pops frames. Every method call performed by the thread is stored in the corresponding run-time stack. The stack area is used to store Method call information, Local variables, Method parameters, Return AddressAfter completing all method calls the stack becomes empty and that empty stack is destroyed by the JVM just before terminating the thread. 4. Program Counter RegisterEach JVM thread that carries out the task of a specific method has a program counter register associated with it.The non-native method has a PC that stores the address of the available JVM instruction whereas, in a native method, the value of the program counter is undefined. PC register is capable of storing the return address or a native pointer on some specific platform.5. Native Method StackAlso called C stacks, native method stacks are not written in Java language. This memory is allocated for each thread when it's created And it can be of a fixed or dynamic nature. Skip to contentHere, we have listed topic-wise the most important 50+ basic Core Java interview questions with the best possible answers for 2024. These basic Java interview questions are often asked from freshers and beginners in any Java technical interview round.We have also covered the most important coding programming questions with answers that will boost up your knowledge to crack the interview round.Basic Java Class and Object Interview Questions1. What is an object in Java?Ans: An object in Java is any real-world thing that has properties and actions. In other words, an entity that has state, behavior, and identity is known as object.2. What are the characteristics of an object in Java?Ans: An object has three characteristics that are as follows:a) State: It represents the properties of an object. It is represented by instance variable/attribute of an object.b) Behavior: It represents functionality or actions. It is represented by methods/functions in Java.c) Identity: It represents the unique name of an object. It differentiates one object from the other. The unique name of an object is used to identify the object.3. What is a class in Java?Ans: A class is a user-defined data type that acts as a template for creating objects of the identical type. It represents the common properties and actions (functions) of an object.4. Why do we need a class in Java?Ans: We need a class for packing together a group of logical related data items (fields) and methods (functions) that work on them.5. What is an attribute?Ans: A variable that is defined in a class is called attribute. 6. What is instantiating of an object?Ans: The process of creating an object of a particular class is called instantiating of an object.7. Why is an object called an instance of a class?Ans: An object is called an instance of a class because every object created from a class gets its own instances of variables defined inside a class. Multiple objects can be constructed from the same class.8. Is a class a data type in Java?Ans: Yes, a class is also considered a user-defined data type because a user defines a class.9. Can a class represent an object?Ans: No, a class never represents an object. It represents data and actions that an object will have.10. Which of the following are valid class names?Student, _Student, class, true, _, CollegeStudent, schoolName, null, _class, Class99Ans: Student, _Student, _, CollegeStudent, _class, Class99 are the valid class names. class, true, null, schoolName are not valid class names.11. What is the difference between class and object in Java?Ans: The differences between class and object are as follows:A class is a user-defined data type, whereas an object is an instance of class data type.A class represents the type of objects, whereas an object represents particular instances of things.Classes do not occupy memory location, but objects occupy memory location.Classes cannot be manipulated due to not available in the memory location, but objects can be manipulated.12. What is the meaning of creating an object in Java?Ans: Creating an object means allocating memory to store the data of variables temporarily. i.e. we create an object of a class to store data temporarily. 13. How to create an object in Java?Ans: In Java, an object of a class is created using the new keyword in three steps. They are as follows.Declaration of a reference variable.Creation of an object.Linking the object and reference variable.14. What is the general syntax to create an object of class?Ans: The general syntax to create an object of a class is as follows:Classname object_reference_variable = new Classname();15. Consider a class whose name is College and object reference variable myCollege. It is declared like this:College myCollege;Answers the following questions:a) Is an object of class College created?b) What is the meaning of this statement: College myCollege;?Ans a. No, an object of class College is not created still. It only refers to an object.Ans b. The statement College myCollege; tells the JVM to allocate memory space for a reference variable and names that reference variable myCollege. The reference variable is of type College.16. What is an object reference in Java?Ans: An object reference is a unique hexadecimal number that represents a memory address of the object. When an object is created, a new reference number is allocated to it. [adinserter block="2″] 17. Where the address of an object is stored in the memory?Ans: The address of an object is stored in the object reference variable in the stack memory.18. What is an object reference variable in Java?Ans: Object reference variable is a variable that stores the address of an object in the stack memory.19. Suppose we have created an object of a class Student like this: new Student();Answers the following questions:a) Is an object of class Student created?b) What is the meaning of this statement: new Student();Ans a: Yes, an object of class Student is created.Ans b: The statement new Student(); tells the JVM to allocate memory space for a new Student object on the heap.20. For which purpose we create an object of class in Java?Ans: We create an object of a class in java to store data temporarily in Java application and to access members of any particular class.21. How many ways to create an object in Java?Ans: There are several ways to create an object of class in Java. They are as follows.Using the new keywordUsing Class.forNameUsing Clone.Using Object Deserialization.Using ClassLoader.22. What is new in Java?Ans: In Java, a new is a special keyword that is used to create an object of the class. It allocates the memory to store an object during runtime and returns a reference to it.23. What is void in Java?Ans: Void is a keyword that indicates that this method does not return any data back to the class of an object. 24. What is the difference between reference variable and variable?Ans: Reference variable holds the address of an object whereas a variable holds data.25. What is object declaration in Java?Ans: The process of defining a variable along with its data type and name is called the declaration of state of an object or simply object declaration.26. What is object initialization in Java?Ans: The process of assigning a value of the variable is called initialization of state of an object or simply object initialization.27. How to initialize the state of object in Java?Ans: There are three ways by which we can initialize the state of an object. In other words, we can initialize value of variables in Java by using three ways. They are as follows:By using constructorBy using a reference variableBy using a method.28. What is an anonymous object in Java?Ans: An object which has no reference variable is called anonymous object in Java29. Write the syntax to create an anonymous object in Java?Ans: The general syntax to create an anonymous object is as follows:new Class_name();30. What is an anonymous class in Java?Ans: A class that does not have a name is called anonymous class in Java.31. When to use an anonymous object?Ans: Anonymous object creation is useful when it is not used more than once.32. Define concrete class in Java.Ans: A class whose object can be created and whose all methods have body is called concrete class.Basic Java Data types and Variables Interview Questions1. What is a statement in Java?Ans: A statement in Java consists of variables, constants, operators, comments, keywords, identifiers, punctuators, etc.2. What are Java tokens?And: Tokens in Java are the various elements in java program that are identified by the compiler. It is the smallest element of a program that is meaningful to the compiler.For example: final double p = 3.14; // constant. This statement consists of six tokens: 'final', 'double', 'p', '=', '3.14', and ';'.3. What are keywords in Java?Ans: Keywords in Java are predefined code (or words) that has a specific meaning and that meaning can be changed. Some commonly used keywords are: class, new, null, void, return, abstract, package, etc.4. Is main keyword in Java?Ans: No, main is not a keyword in Java.5. What is data type in Java?Ans: A data type in java is a term that specifies memory size and type of values that can be stored into the memory location. In other words, a data type defines different values that a variable can take.6. What are the types of data types in Java?Ans: There are two types of data types in java that are as follows:Primitive data typesNon-primitive data types7. What is a Primitive data type in Java?Ans: A data type whose variable can store only one value at a time is called primitive data type. Primitive data types are predefined in Java and cannot be developed by programmers. It can be used by programmers when creating variables in the program.8. What are the different types of primitive data types supported by Java programming language?Ans: Java defines eight primitive data types: boolean, char, byte, short, int, long, char, float, and double.9. What are the different types of Integer data types?Ans: Java defines four integer data types: byte, short, int, and long.10. Which is the smallest integer data type?Ans: Byte is the smallest integer data type that has the least memory size allocated.11. What is the memory size of byte data type?Ans: The memory size of a byte is 8 bits.12. What is the default value of byte?Ans: The default value of byte is 0.13. What is the range value of byte data type?Ans: The range value of byte is from -128 (2^7) to 127 (inclusive)(2^7 – 1).14. What is the storage capacity of short integer data type?Ans: The default memory size (storage capacity) of short is 16 bits (i.e. 4 bytes).15. What is the default value of short?Ans: The default value of short is 0.16. Which data type is mostly used for integer values in Java programming?Ans: int data type.17. What are the memory size and default value of int data type?Ans: The memory size of int is 32 bits (4 bytes). The default value of int is 0.18. What is the default memory size and value of long data type?Ans: The default memory size of long data type is 64 bits (8 bytes) and default value is 0.19. Will the code compile successfully?public class Test { public static void main(String[] args) { byte num = 129; System.out.println(num); } }Output:Compile-time error: Type mismatch: cannot convert from int to byteThis error occurs because the value is out of the range of byte type. The range of byte is -128 to +127.20. Will the code compile successfully? If yes, what will be the output of the below program?public class Test { public static void main(String[] args) { short num = 12_9_10; System.out.println(num); } }Answer: Yes, the code will be compiled successfully. The output will be 12910.21. Which of the following lists of integer (numeric) primitive types is presented in order from smallest to largest data type?a) byte, short, int, longb) int, short, byte, longc) short, byte, int, longd) short, int, byte, longAns: a (byte, short, int, long)22. Suppose we declare int x = 10; Is this declaration advisable?Ans: No, it is not advisable. We should use a byte variable instead of using int variable because byte will take less memory size and improve the speed of execution of the program.Whenever possible, you should use smaller data types because wider data types take more time for manipulation.23. What will be the output of the following program?public class Test { public static void main(String[] args) { long x = (long) 200000.255; System.out.println(x); } }Output: 20000024. What is the use of float and double data types in Java programming language?Ans: Float data type is used to represent the decimal number which can hold 6 to 7 decimal digits. Double data type is also used to represent decimal numbers up to 15 decimal digits accurately.25. What are the default memory size and default value of float data type?Ans: The default memory size allotted for float data type is 32 bits (i.e. 4 bytes). The default value for float data type is 0.0f.26. What are the default memory size and default value of double data type?Ans: The default memory size allotted for double is 64 bits (i.e. 8 bytes). The default value is 0.0d.27. What is the use of char data type in Java programming language?Ans: A char data type is mainly used to store a single character like P, a, b, z, x, etc.28. What are the default memory size and default value of char data type?Ans: The default memory size allotted for char data type is 2 bytes. The default value is u0000.29. What is the use of boolean data type in Java?Ans: Boolean data type is generally used to test a particular conditional statement during the execution of the program. It represents one bit of information as either true or false.30. What is the default memory size and default value for boolean data type in Java?Ans: The default memory size allotted for boolean data type is 1 bit. The default value is false.31. Why Java takes 2 bytes of memory for storing character?Ans: Java supports more than 18 international languages. To handle more than 18 international languages, 1 byte of memory is not sufficient for storing all characters and symbols present in 18 languages.32. Why take boolean data types zero bytes of memory?Ans: Boolean data type takes zero bytes of memory space because boolean data type in Java is implemented by Sun Micro System using the concept of a flip-flop.A flip-flop is a general-purpose register that stores one bit of information (one for true and zero for false).33. Which of the following lists of primitive data types are presented in order from largest to smallest data type?a) byte, char, float, doubleb) byte, char, double, floatc) double, float, byte, chard) char, double, float, intAns: c (double, float, byte, char)34. Which of the following declarations will give compile-time error?a) double d1 = 5.50f;b) double d2 = 5.0;c) float f1 = 10f;d) float f2 = 10.00;Ans: d35. Which of the following declarations will not compile?a) int num1 = 999;b) int num2 = _999;c) int num3 = 9_9_9;d) All of the above compileAns: b36. Which of the following declarations will give compile time error?a) double x, int y = 0;b) int x, y;c) int x, y = 0;d) int x = 0, y = 0;Ans: a37. What is the difference between byte and char data types in Java?Ans: The main difference between byte and char is that a byte can store raw binary data whereas a char stores characters or text data.38. What will happen when we assign a variable of primitive data type to another variable of same type?Ans: When a variable of primitive data type is assigned to another variable, a copy is created.39. What is non-primitive data type in Java?Ans: A data type that is used to store a group of values (either of same type or different type or both) is called non primitive data type or user-defined data type. It is created by programmers in their program. A non-primitive data type is also known as advanced data types in java.40. Why non-primitive data type is also called reference data type in Java?Ans: When we declare a variable of non-primitive data type, it references a memory location where data is stored in the heap memory. Therefore, the variable of a non-primitive data type is also called referenced data type in Java.41. What is the default value of any reference variable?Ans: null42. What are types of non-primitive data types in Java?Ans: There are five types of non-primitive data types in Java. They are:ClassObjectStringArrayInterface43. What is the difference between primitive data types and non-primitive data types in Java?Ans: Refer to this tutorial: Non-primitive data types in Java44. In which memory location JVM stores primitive type values?Ans: Stack memory.45. What will happen when we assign a variable of reference data type to another variable of same reference type?Ans: When a variable of reference type is assigned to another reference variable, both points to the same object.46. If a value to variable of reference data type is not assigned, what does it contain?Ans: It contains a null default value when it is unassigned.47. What is an identifier in Java?Ans: An identifier is a sequence of characters that are made up of letters, digits, underscores(_), and dollar signs ($). It represents a name that identifies elements such as variables, methods, classes, and other items in the program. For example, Student, num1, num2, main, etc.48. Which of the following identifiers are valid?student, Test, area, a++, –a, 5#R, $5, #77, class, public, int, x, y, radius.Ans: Test, student, area, radius, $5, x, y, radius are valid identifiers because they follow rules.Related Java Interview Questions on Topic WiseIn this tutorial, we have covered almost all the important basic Core Java interview questions and answers for 2025. Hope that you will have understood all the answers to the basic Java interview questions. All the best!!!Tags# Basic Java Interview Questions and Answers