30 000 слов по английскому

МГ-101 Тимофеев Д.А.

СОДЕРЖАНИЕ:

1. Java Interview Questions

Previous Page

Next Page

Dear readers, these **Java Interview Questions** have been designed especially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **Java Programming Language**. As per my experience, good interviewers hardly planned to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer −

What do you know about Java?

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

What are the supported platforms by Java Programming Language?

Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX/Linux like HP-Unix, Sun Solaris, Redhat Linux, Ubuntu, CentOS, etc.

List any five features of Java?

Some features include Object Oriented, Platform Independent, Robust, Interpreted, Multi-threaded

Why is Java Architectural Neutral?

It's compiler generates an architecture-neutral object file format, which makes the compiled code to be executable on many processors, with the presence of Java runtime system.

How Java enabled High Performance?

Java uses Just-In-Time compiler to enable high performance. Just-In-Time compiler is a program that turns Java bytecode, which is a program that contains instructions that must be interpreted into instructions that can be sent directly to the processor.

Why Java is considered dynamic?

It is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

What is Java Virtual Machine and how it is considered in context of Java's platform independent feature?

When Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.

List two Java IDE's?

Netbeans, Eclipse, etc.

List some Java keywords(unlike C, C++ keywords)?

Some Java keywords are import, super, finally, etc.

What do you mean by Object?

Object is a runtime entity and it’s state is stored in fields and behavior is shown via methods. Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication.

Define class?

A class is a blue print from which individual objects are created. A class can contain fields and methods to describe the behavior of an object.

What kind of variables a class can consist of?

A class consist of Local variable, instance variables and class variables.

What is a Local Variable?

Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and it will be destroyed when the method has completed.

What is a Instance Variable?

Instance variables are variables within a class but outside any method. These variables are instantiated when the class is loaded.

What is a Class Variable?

These are variables declared with in a class, outside any method, with the static keyword.

What is Singleton class?

Singleton class control object creation, limiting the number to one but allowing the flexibility to create more objects if the situation changes.

What do you mean by Constructor?

Constructor gets invoked when a new object is created. Every class has a constructor. If we do not explicitly write a constructor for a class the java compiler builds a default constructor for that class.

List the three steps for creating an Object for a class?

An Object is first declared, then instantiated and then it is initialized.

What is the default value of byte datatype in Java?

Default value of byte datatype is 0.

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

Default value of float and double datatype in different as compared to C/C++. For float its 0.0f and for double it’s 0.0d

When a byte datatype is used?

This data type is used to save space in large arrays, mainly in place of integers, since a byte is four times smaller than an int.

What is a static variable?

Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block.

What do you mean by Access Modifier?

Java provides access modifiers to set access levels for classes, variables, methods and constructors. A member has package or default accessibility when no accessibility modifier is specified.

What is protected access modifier?

Variables, methods and constructors which are declared protected in a superclass can be accessed only by the subclasses in other package or any class within the package of the protected members' class.

What do you mean by synchronized Non Access Modifier?

Java provides these modifiers for providing functionalities other than Access Modifiers, synchronized used to indicate that a method can be accessed by only one thread at a time.

According to Java Operator precedence, which operator is considered to be with highest precedence?

Postfix operators i.e () [] . is at the highest precedence.

Variables used in a switch statement can be used with which datatypes?

Variables used in a switch statement can only be a string, enum, byte, short, int, or char.

When parseInt() method can be used?

This method is used to get the primitive data type of a certain String.

Why is String class considered immutable?

The String class is immutable, so that once it is created a String object cannot be changed. Since String is immutable it can safely be shared between many threads ,which is considered very important for multithreaded programming.

Why is StringBuffer called mutable?

The String class is considered as immutable, so that once it is created a String object cannot be changed. If there is a necessity to make alot of modifications to Strings of characters then StringBuffer should be used.

What is the difference between StringBuffer and StringBuilder class?

Use StringBuilder whenever possible because it is faster than StringBuffer. But, if thread safety is necessary then use StringBuffer objects.

Which package is used for pattern matching with regular expressions?

java.util.regex package is used for this purpose.

java.util.regex consists of which classes?

java.util.regex consists of three classes − Pattern class, Matcher class and PatternSyntaxException class.

What is finalize() method?

It is possible to define a method that will be called just before an object's final destruction by the garbage collector. This method is called finalize( ), and it can be used to ensure that an object terminates cleanly.

What is an Exception?

An exception is a problem that arises during the execution of a program. Exceptions are caught by handlers positioned along the thread's method invocation stack.

What do you mean by Checked Exceptions?

It is an exception that is typically a user error or a problem that cannot be foreseen by the programmer. For example, if a file is to be opened, but the file cannot be found, an exception occurs. These exceptions cannot simply be ignored at the time of compilation.

Explain Runtime Exceptions?

It is an exception that occurs that probably could have been avoided by the programmer. As opposed to checked exceptions, runtime exceptions are ignored at the time of compliation.

Which are the two subclasses under Exception class?

The Exception class has two main subclasses : IOException class and RuntimeException Class.

When throws keyword is used?

If a method does not handle a checked exception, the method must declare it using the throwskeyword. The throws keyword appears at the end of a method's signature.

When throw keyword is used?

An exception can be thrown, either a newly instantiated one or an exception that you just caught, by using throw keyword.

How finally used under Exception Handling?

The finally keyword is used to create a block of code that follows a try block. A finally block of code always executes, whether or not an exception has occurred.

What things should be kept in mind while creating your own exceptions in Java?

While creating your own exception −

- All exceptions must be a child of Throwable.
- If you want to write a checked exception that is automatically enforced by the Handle or Declare Rule, you need to extend the Exception class.
- You want to write a runtime exception, you need to extend the RuntimeException class.

Define Inheritance?

It is the process where one object acquires the properties of another. With the use of inheritance the information is made manageable in a hierarchical order.

When super keyword is used?

If the method overrides one of its superclass's methods, overridden method can be invoked through the use of the keyword super. It can be also used to refer to a hidden field.

What is Polymorphism?

Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.

What is Abstraction?

It refers to the ability to make a class abstract in OOP. It helps to reduce the complexity and also improves the maintainability of the system.

What is Abstract class?

These classes cannot be instantiated and are either partially implemented or not at all implemented. This class contains one or more abstract methods which are simply method declarations without a body.

When Abstract methods are used?

If you want a class to contain a particular method but you want the actual implementation of that method to be determined by child classes, you can declare the method in the parent class as abstract.

What is Encapsulation?

It is the technique of making the fields in a class private and providing access to the fields via public methods. If a field is declared private, it cannot be accessed by anyone outside the class, thereby hiding the fields within the class. Therefore encapsulation is also referred to as data hiding.

What is the primary benefit of Encapsulation?

The main benefit of encapsulation is the ability to modify our implemented code without breaking the code of others who use our code. With this Encapsulation gives maintainability, flexibility and extensibility to our code.

What is an Interface?

An interface is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

Give some features of Interface?

It includes −

- Interface cannot be instantiated
- An interface does not contain any constructors.
- All of the methods in an interface are abstract.

Define Packages in Java?

A Package can be defined as a grouping of related types(classes, interfaces, enumerations and annotations) providing access protection and name space management.

Why Packages are used?

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

What do you mean by Multithreaded program?

A multithreaded program contains two or more parts that can run concurrently. Each part of such a program is called a thread, and each thread defines a separate path of execution.

What are the two ways in which Thread can be created?

Thread can be created by: implementing Runnable interface, extending the Thread class.

What is an applet?

An applet is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal.

An applet extend which class?

An applet extends java.applet.Applet class.

Explain garbage collection in Java?

It uses garbage collection to free the memory. By cleaning those objects that is no longer reference by any of the program.

Define immutable object?

An immutable object can't be changed once it is created.

Explain the usage of this() with constructors?

It is used with variables or methods and used to call constructer of same class.

Explain Set Interface?

It is a collection of element which cannot contain duplicate elements. The Set interface contains only methods inherited from Collection and adds the restriction that duplicate elements are prohibited.

Explain TreeSet?

It is a Set implemented when we want elements in a sorted order.

What is Comparable Interface?

It is used to sort collections and arrays of objects using the collections.sort() and java.utils. The objects of the class implementing the Comparable interface can be ordered.

Difference between throw and throws?

It includes:

- Throw is used to trigger an exception where as throws is used in declaration of exception.
- Without throws, Checked exception cannot be handled where as checked exception can be propagated with throws.

Explain the following line used under Java Program −

public static void main (String args[ ])

The following shows the explanation individually −

- public − it is the access specifier.
- static − it allows main() to be called without instantiating a particular instance of a class.
- void − it affirns the compiler that no value is returned by main().
- main() − this method is called at the beginning of a Java program.
- String args[ ] − args parameter is an instance array of class String

Define JRE i.e. Java Runtime Environment?

Java Runtime Environment is an implementation of the Java Virtual Machine which executes Java programs. It provides the minimum requirements for executing a Java application;

What is JAR file?

JAR files is Java Archive fles and it aggregates many files into one. It holds Java classes in a library. JAR files are built on ZIP file format and have .jar file extension.

What is a WAR file?

This is Web Archive File and used to store XML, java classes, and JavaServer pages. which is used to distribute a collection of JavaServer Pages, Java Servlets, Java classes, XML files, static Web pages etc.

Define JIT compiler?

It improves the runtime performance of computer programs based on bytecode.

What is the difference between object oriented programming language and object based programming language?

Object based programming languages follow all the features of OOPs except Inheritance. JavaScript is an example of object based programming languages.

What is the purpose of default constructor?

The java compiler creates a default constructor only if there is no constructor in the class.

Can a constructor be made final?

No, this is not possible.

What is static block?

It is used to initialize the static data member, It is excuted before main method at the time of classloading.

Define composition?

Holding the reference of the other class within some other class is known as composition.

What is function overloading?

If a class has multiple functions by same name but different parameters, it is known as Method Overloading.

What is function overriding?

If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding.

Difference between Overloading and Overriding?

Method overloading increases the readability of the program. Method overriding provides the specific implementation of the method that is already provided by its super class parameter must be different in case of overloading, parameter must be same in case of overriding.

What is final class?

Final classes are created so the methods implemented by that class cannot be overridden. It can’t be inherited.

What is NullPointerException?

A NullPointerException is thrown when calling the instance method of a null object, accessing or modifying the field of a null object etc.

What are the ways in which a thread can enter the waiting state?

A thread can enter the waiting state by invoking its sleep() method, by blocking on IO, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

How does multi-threading take place on a computer with a single CPU?

The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

What invokes a thread's run() method?

After a thread is started, via its start() method of the Thread class, the JVM invokes the thread's run() method when the thread is initially executed.

Does it matter in what order catch statements for FileNotFoundException and IOException are written?

Yes, it does. The FileNoFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

What is the difference between yielding and sleeping?

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

Why Vector class is used?

The Vector class provides the capability to implement a growable array of objects. Vector proves to be very useful if you don't know the size of the array in advance, or you just need one that can change sizes over the lifetime of a program.

How many bits are used to represent Unicode, ASCII, UTF-16, and UTF-8 characters?

Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

What are Wrapper classes?

These are classes that allow primitive types to be accessed as objects. Example: Integer, Character, Double, Boolean etc.

What is the difference between a Window and a Frame?

The Frame class extends Window to define a main application window that can have a menu bar.

Which package has light weight components?

javax.Swing package. All components in Swing, except JApplet, JDialog, JFrame and JWindow are lightweight components.

What is the difference between the paint() and repaint() methods?

The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.

What is the purpose of File class?

It is used to create objects that provide access to the files and directories of a local file system.

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

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

Which class should you use to obtain design information about an object?

The Class class is used to obtain information about an object's design and java.lang.Class class instance represent classes, interfaces in a running Java application.

What is the difference between static and non-static variables?

A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

What is Serialization and deserialization?

Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

What are use cases?

It is part of the analysis of a program and describes a situation that a program might encounter and what behavior the program should exhibit in that circumstance.

Explain the use of sublass in a Java program?

Sub class inherits all the public and protected methods and the implementation. It also inherits all the default modifier methods and their implementation.

How to add menushortcut to menu item?

If there is a button instance called b1, you may add menu short cut by calling b1.setMnemonic('F'), so the user may be able to use Alt+F to click the button.

Can you write a Java class that could be used both as an applet as well as an application?

Yes, just add a main() method to the applet.

What is the difference between Swing and AWT components?

AWT components are heavy-weight, whereas Swing components are lightweight. Heavy weight components depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button.

What's the difference between constructors and other methods?

Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.

Is there any limitation of using Inheritance?

Yes, since inheritance inherits everything from the super class and interface, it may make the subclass too clustering and sometimes error-prone when dynamic overriding or dynamic overloading in some situation.

When is the ArrayStoreException thrown?

When copying elements between different arrays, if the source or destination arguments are not arrays or their types are not compatible, an ArrayStoreException will be thrown.

Can you call one constructor from another if a class has multiple constructors?

Yes, use this() syntax.

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

The code sleep(2000); puts thread aside for exactly two seconds. The code wait(2000), causes a wait of up to two second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

When ArithmeticException is thrown?

The ArithmeticException is thrown when integer is divided by zero or taking the remainder of a number by zero. It is never thrown in floating-point operations.

What is a transient variable?

A transient variable is a variable that may not be serialized during Serialization and which is initialized by its default value during de-serialization,

What is synchronization?

Synchronization is the capability to control the access of multiple threads to shared resources. synchronized keyword in java provides locking which ensures mutual exclusive access of shared resource and prevent data race.

What is the Collections API?

The Collections API is a set of classes and interfaces that support operations on collections of objects.

Does garbage collection guarantee that a program will not run out of memory?

Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection.

The immediate superclass of the Applet class?

Panel is the immediate superclass. A panel provides space in which an application can attach any other component, including other panels.

Which Java operator is right associative?

The = operator is right associative.

What is the difference between a break statement and a continue statement?

A break statement results in the termination of the statement to which it applies (switch, for, do, or while). A continue statement is used to end the current loop iteration and return control to the loop statement.

If a variable is declared as private, where may the variable be accessed?

A private variable may only be accessed within the class in which it is declared.

What is the purpose of the System class?

The purpose of the System class is to provide access to system resources.

List primitive Java types?

The eight primitive types are byte, char, short, int, long, float, double, and boolean.

What is the relationship between clipping and repainting under AWT?

When a window is repainted by the AWT painting thread, it sets the clipping regions to the area of the window that requires repainting.

Which class is the immediate superclass of the Container class?

Component class is the immediate super class.

What class of exceptions are generated by the Java run-time system?

The Java runtime system generates RuntimeException and Error exceptions.

Under what conditions is an object's finalize() method invoked by the garbage collector?

The garbage collector invokes an object's finalize() method when it detects that the object has become unreachable.

How can a dead thread be restarted?

A dead thread cannot be restarted.

Which arithmetic operations can result in the throwing of an ArithmeticException?

Integer / and % can result in the throwing of an ArithmeticException.

Variable of the boolean type is automatically initialized as?

The default value of the boolean type is false.

Can try statements be nested?

Yes

What are ClassLoaders?

A class loader is an object that is responsible for loading classes. The class ClassLoader is an abstract class.

What is the difference between an Interface and an Abstract class?

An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation.

What will happen if static modifier is removed from the signature of the main method?

Program throws "NoSuchMethodError" error at runtime.

What is the default value of an object reference declared as an instance variable?

Null, unless it is defined explicitly.

Can a top level class be private or protected?

No, a top level class can not be private or protected. It can have either "public" or no modifier.

Why do we need wrapper classes?

We can pass them around as method parameters where a method expects an object. It also provides utility methods.

What is the difference between error and an exception?

An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. Exceptions are conditions that occur because of bad input etc. e.g. FileNotFoundException will be thrown if the specified file does not exist.

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

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block or a finally block.

When a thread is created and started, what is its initial state?

A thread is in the ready state as initial state after it has been created and started.

What is the Locale class?

The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region.

What are synchronized methods and synchronized statements?

Synchronized methods are methods that are used to control access to an object. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

What is runtime polymorphism or dynamic method dispatch?

Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time. In this process, an overridden method is called through the reference variable of a superclass.

What is Dynamic Binding(late binding)?

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run-time.

Can constructor be inherited?

No, constructor cannot be inherited.

What are the advantages of ArrayList over arrays?

ArrayList can grow dynamically and provides more powerful insertion and search mechanisms than arrays.

Why deletion in LinkedList is fast than ArrayList?

Deletion in linked list is fast because it involves only updating the next pointer in the node before the deleted node and updating the previous pointer in the node after the deleted node.

How do you decide when to use ArrayList and LinkedList?

If you need to frequently add and remove elements from the middle of the list and only access the list elements sequentially, then LinkedList should be used. If you need to support random access, without inserting or removing elements from any place other than the end, then ArrayList should be used.

What is a Values Collection View ?

It is a collection returned by the values() method of the Map Interface, It contains all the objects present as values in the map.

What is dot operator?

The dot operator(.) is used to access the instance variables and methods of class objects.It is also used to access classes and sub-packages from a package.

Where and how can you use a private constructor?

Private constructor is used if you do not want other classes to instantiate the object and to prevent subclassing.T

What is type casting?

Type casting means treating a variable of one type as though it is another type.

Describe life cycle of thread?

A thread is a execution in a program. The life cycle of a thread include −

- Newborn state
- Runnable state
- Running state
- Blocked state
- Dead state

What is the difference between the >> and >>> operators?

The >> operator carries the sign bit when shifting right. The >>> zero-fills bits that have been shifted out.

Which method of the Component class is used to set the position and size of a component?

setBounds() method is used for this purpose.

What is the range of the short type?

The range of the short type is -(2^15) to 2^15 - 1.

What is the immediate superclass of Menu?

MenuItem class

Does Java allow Default Arguments?

No, Java does not allow Default Arguments.

Which number is denoted by leading zero in java?

Octal Numbers are denoted by leading zero in java, example: 06

Which number is denoted by leading 0x or 0X in java?

Hexadecimal Numbers are denoted by leading 0x or 0X in java, example − 0XF

Break statement can be used as labels in Java?

Yes, an example can be *break one;*

Where import statement is used in a Java program?

Import statement is allowed at the beginning of the program file after package statement.

Explain suspend() method under Thread class>

It is used to pause or temporarily stop the execution of the thread.

Explain isAlive() method under Thread class?

It is used to find out whether a thread is still running or not.

What is currentThread()?

It is a public static method used to obtain a reference to the current thread.

Explain main thread under Thread class execution?

The main thread is created automatically and it begins to execute immediately when a program starts. It ia thread from which all other child threads originate.

Life cycle of an applet includes which steps?

Life cycle involves the following steps −

- Initialization
- Starting
- Stopping
- Destroying
- Painting

Why is the role of init() method under applets?

It initializes the applet and is the first method to be called.

Which method is called by Applet class to load an image?

getImage(URL object, filename) is used for this purpose.

Define code as an attribute of Applet?

It is used to specify the name of the applet class.

Define canvas?

It is a simple drawing surface which are used for painting images or to perform other graphical operations.

Define Network Programming?

It refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network.

What is a Socket?

Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

Advantages of Java Sockets?

Sockets are flexible and sufficient. Efficient socket based programming can be easily implemented for general communications. It cause low network traffic.

Disadvantages of Java Sockets?

Socket based communications allows only to send packets of raw data between applications. Both the client-side and server-side have to provide mechanisms to make the data useful in any way.

Which class is used by server applications to obtain a port and listen for client requests?

java.net.ServerSocket class is used by server applications to obtain a port and listen for client requests

Which class represents the socket that both the client and server use to communicate with each other?

java.net.Socket class represents the socket that both the client and server use to communicate with each other.

Why Generics are used in Java?

Generics provide compile-time type safety that allows programmers to catch invalid types at compile time. Java Generic methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods or, with a single class declaration, a set of related types.

What environment variables do I need to set on my machine in order to be able to run Java programs?

CLASSPATH and PATH are the two variables.

Is there any need to import java.lang package?

No, there is no need to import this package. It is by default loaded internally by the JVM.

What is Nested top-level class?

If a class is declared within a class and specify the static modifier, the compiler treats the class just like any other top-level class. Nested top-level class is an Inner class.

What is Externalizable interface?

Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism.

If System.exit (0); is written at the end of the try block, will the finally block still execute?

No in this case the finally block will not execute because when you say System.exit (0); the control immediately goes out of the program, and thus finally never executes.

What is daemon thread?

Daemon thread is a low priority thread, which runs intermittently in the back ground doing the garbage collection operation for the java runtime system.

Which method is used to create the daemon thread?

setDaemon method is used to create a daemon thread.

Which method must be implemented by all threads?

All tasks must implement the run() method

What is the GregorianCalendar class?

The GregorianCalendar provides support for traditional Western calendars

What is the SimpleTimeZone class?

The SimpleTimeZone class provides support for a Gregorian calendar .

What is the difference between the size and capacity of a Vector?

The size is the number of elements actually stored in the vector, while capacity is the maximum number of elements it can store at a given instance of time.

Can a vector contain heterogenous objects?

Yes a Vector can contain heterogenous objects. Because a Vector stores everything in terms of Object.

What is an enumeration?

An enumeration is an interface containing methods for accessing the underlying data structure from which the enumeration is obtained. It allows sequential access to all the elements stored in the collection.

What is difference between Path and Classpath?

Path and Classpath are operating system level environment variales. Path is defines where the system can find the executables(.exe) files and classpath is used to specify the location of .class files.

Can a class declared as private be accessed outside it's package?

No, it's not possible to accessed outside it's package.

What are the restriction imposed on a static method or a static block of code?

A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

Can an Interface extend another Interface?

Yes an Interface can inherit another Interface, for that matter an Interface can extend more than one Interface.

Which object oriented Concept is achieved by using overloading and overriding?

Polymorphism

What is an object's lock and which object's have locks?

An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock.

What is Downcasting?

It is the casting from a general to a more specific type, i.e. casting down the hierarchy.

What are order of precedence and associativity and how are they used?

Order of precedence determines the order in which operators are evaluated in expressions. Associatity determines whether an expression is evaluated left-to-right or right-to-left.

If a method is declared as protected, where may the method be accessed?

A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

What is the difference between inner class and nested class?

When a class is defined within a scope of another class, then it becomes inner class. If the access modifier of the inner class is static, then it becomes nested class.

What restrictions are placed on method overriding?

Overridden methods must have the same name, argument list, and return type. The overriding method may not limit the access of the method it overrides.

What is constructor chaining and how is it achieved in Java?

A child object constructor always first needs to construct its parent. In Java it is done via an implicit call to the no-args constructor as the first statement.

Can a double value be cast to a byte?

Yes, a double value can be cast to a byte.

How does a try statement determine which catch clause should be used to handle an exception?

When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

What will be the default values of all the elements of an array defined as an instance variable?

If the array is an array of primitive types, then all the elements of the array will be initialized to the default value corresponding to that primitive type.

What is Next?

Further, you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

2. Java - Multithreading

Previous Page

Next Page

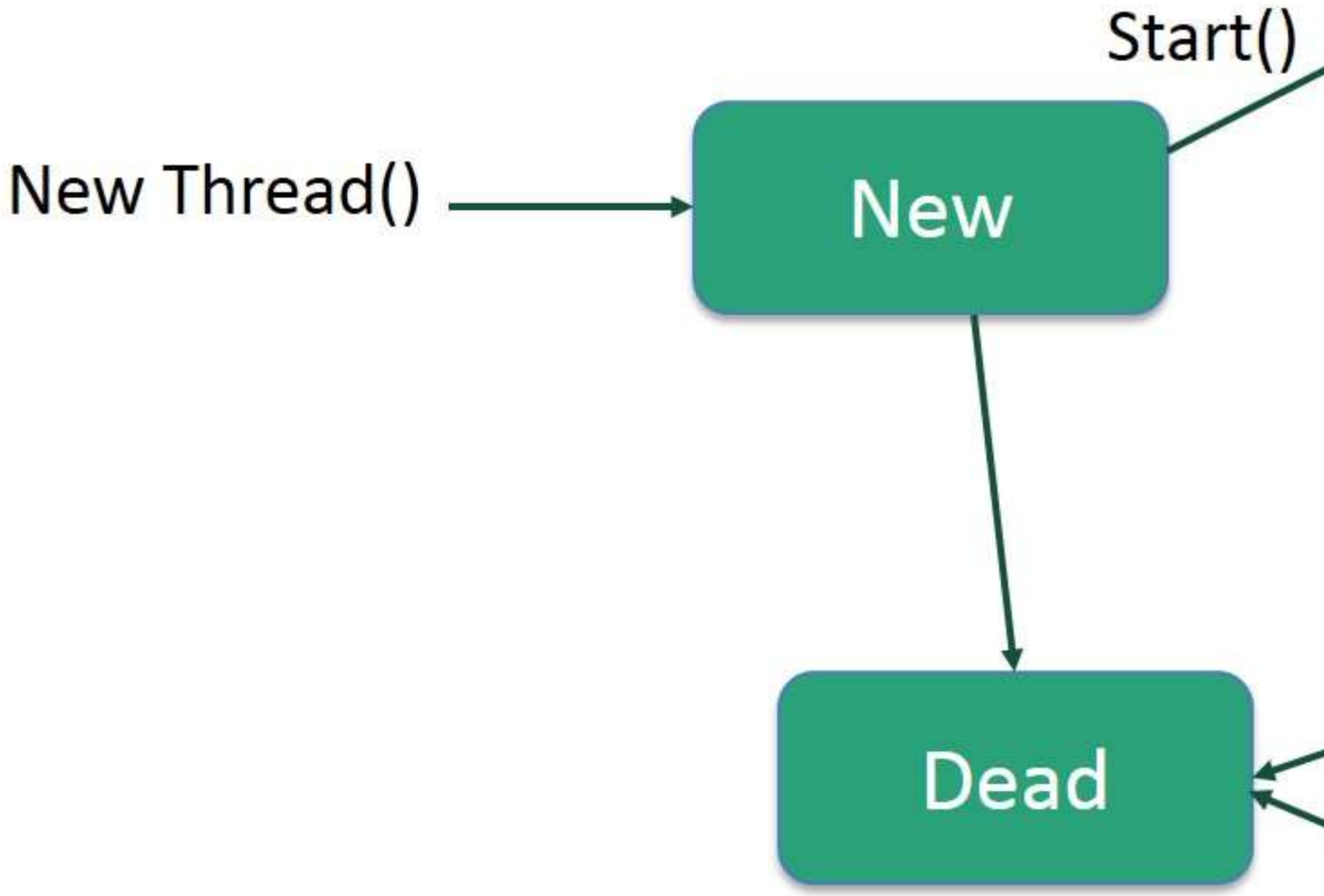
Java is a *multi-threaded programming language* which means we can develop multi-threaded program using Java. A multi-threaded program contains two or more parts that can run concurrently and each part can handle a different task at the same time making optimal use of the available resources specially when your computer has multiple CPUs.

By definition, multitasking is when multiple processes share common processing resources such as a CPU. Multi-threading extends the idea of multitasking into applications where you can subdivide specific operations within a single application into individual threads. Each of the threads can run in parallel. The OS divides processing time not only among different applications, but also among each thread within an application.

Multi-threading enables you to write in a way where multiple activities can proceed concurrently in the same program.

Life Cycle of a Thread

A thread goes through various stages in its life cycle. For example, a thread is born, started, runs, and then dies. The following diagram shows the complete life cycle of a thread.



Following are the stages of the life cycle −

- **New** − A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread. It is also referred to as a **born thread**.
- **Runnable** − After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be executing its task.
- **Waiting** − Sometimes, a thread transitions to the waiting state while the thread waits for another thread to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.
- **Timed Waiting** − A runnable thread can enter the timed waiting state for a specified interval of time. A thread in this state transitions back to the runnable state when that time interval expires or when the event it is waiting for occurs.
- **Terminated (Dead)** − A runnable thread enters the terminated state when it completes its task or otherwise terminates.

Thread Priorities

Every Java thread has a priority that helps the operating system determine the order in which threads are scheduled.

Java thread priorities are in the range between MIN_PRIORITY (a constant of 1) and MAX_PRIORITY (a constant of 10). By default, every thread is given priority NORM_PRIORITY (a constant of 5).

Threads with higher priority are more important to a program and should be allocated processor time before lower-priority threads. However, thread priorities cannot guarantee the order in which threads execute and are very much platform dependent.

Create a Thread by Implementing a Runnable Interface

If your class is intended to be executed as a thread then you can achieve this by implementing a **Runnable** interface. You will need to follow three basic steps −

Step 1

As a first step, you need to implement a run() method provided by a **Runnable** interface. This method provides an entry point for the thread and you will put your complete business logic inside this method. Following is a simple syntax of the run() method −

```
public void run( )
```

Step 2

As a second step, you will instantiate a **Thread** object using the following constructor −

```
Thread(Runnable threadObj, String threadName);
```

Where, *threadObj* is an instance of a class that implements the **Runnable** interface and **threadName** is the name given to the new thread.

Step 3

Once a Thread object is created, you can start it by calling **start()** method, which executes a call to run( ) method. Following is a simple syntax of start() method −

```
void start();
```

Example

Here is an example that creates a new thread and starts running it −

Live Demo

```
class RunnableDemo implements Runnable {
  private Thread t;
  private String threadName;

  RunnableDemo( String name) {
    threadName = name;
    System.out.println("Creating " +  threadName );
  }

  public void run() {
    System.out.println("Running " +  threadName );
    try {
      for(int i = 4; i > 0; i--) {
        System.out.println("Thread: " + threadName + ", " + i);
```

```
            // Let the thread sleep for a while.
            Thread.sleep(50);
         }
      } catch (InterruptedException e) {
         System.out.println("Thread " +  threadName + " interrupted.");
      }
      System.out.println("Thread " +  threadName + " exiting.");
   }

   public void start () {
      System.out.println("Starting " +  threadName );
      if (t == null) {
         t = new Thread (this, threadName);
         t.start ();
      }
   }
}

public class TestThread {

   public static void main(String args[]) {
      RunnableDemo R1 = new RunnableDemo( "Thread-1");
      R1.start();

      RunnableDemo R2 = new RunnableDemo( "Thread-2");
      R2.start();
   }
}
```

This will produce the following result −

Output

```
Creating Thread-1
Starting Thread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Thread: Thread-1, 4
Running Thread-2
Thread: Thread-2, 4
Thread: Thread-1, 3
Thread: Thread-2, 3
Thread: Thread-1, 2
Thread: Thread-2, 2
Thread: Thread-1, 1
Thread: Thread-2, 1
Thread Thread-1 exiting.
Thread Thread-2 exiting.
```

Create a Thread by Extending a Thread Class

The second way to create a thread is to create a new class that extends **Thread** class using the following two simple steps. This approach provides more flexibility in handling multiple threads created using available methods in Thread class.

Step 1

You will need to override **run( )** method available in Thread class. This method provides an entry point for the thread and you will put your complete business logic inside this method. Following is a simple syntax of run() method −

```
public void run( )
```

Step 2

Once Thread object is created, you can start it by calling **start()** method, which executes a call to run( ) method. Following is a simple syntax of start() method −

```
void start( );
```

Example

Here is the preceding program rewritten to extend the Thread −

Live Demo

```
class ThreadDemo extends Thread {
   private Thread t;
   private String threadName;

   ThreadDemo( String name) {
      threadName = name;
      System.out.println("Creating " +  threadName );
   }

   public void run() {
      System.out.println("Running " +  threadName );
      try {
         for(int i = 4; i > 0; i--) {
            System.out.println("Thread: " + threadName + ", " + i);
            // Let the thread sleep for a while.
            Thread.sleep(50);
         }
      } catch (InterruptedException e) {
         System.out.println("Thread " +  threadName + " interrupted.");
      }
      System.out.println("Thread " +  threadName + " exiting.");
   }

   public void start () {
      System.out.println("Starting " +  threadName );
      if (t == null) {
         t = new Thread (this, threadName);
         t.start ();
      }
   }
}

public class TestThread {

   public static void main(String args[]) {
      ThreadDemo T1 = new ThreadDemo( "Thread-1");
      T1.start();
```

```
    ThreadDemo T2 = new ThreadDemo( "Thread-2");
    T2.start();
  }
}
```

This will produce the following result −

Output

```
Creating Thread-1
Starting Thread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Thread: Thread-1, 4
Running Thread-2
Thread: Thread-2, 4
Thread: Thread-1, 3
Thread: Thread-2, 3
Thread: Thread-1, 2
Thread: Thread-2, 2
Thread: Thread-1, 1
Thread: Thread-2, 1
Thread Thread-1 exiting.
Thread Thread-2 exiting.
```

Thread Methods

Following is the list of important methods available in the Thread class.

Sr.No.	Method & Description
1	**public void start()** Starts the thread in a separate path of execution, then invokes the run() method on this Thread object.
2	**public void run()** If this Thread object was instantiated using a separate Runnable target, the run() method is invoked on that Runnable object.
3	**public final void setName(String name)** Changes the name of the Thread object. There is also a getName() method for retrieving the name.
4	**public final void setPriority(int priority)** Sets the priority of this Thread object. The possible values are between 1 and 10.

5	**public final void setDaemon(boolean on)** A parameter of true denotes this Thread as a daemon thread.
6	**public final void join(long millisec)** The current thread invokes this method on a second thread, causing the current thread to block until the second thread terminates or the specified number of milliseconds passes.
7	**public void interrupt()** Interrupts this thread, causing it to continue execution if it was blocked for any reason.
8	**public final boolean isAlive()** Returns true if the thread is alive, which is any time after the thread has been started but before it runs to completion.

The previous methods are invoked on a particular Thread object. The following methods in the Thread class are static. Invoking one of the static methods performs the operation on the currently running thread.

Sr.No.	Method & Description
1	**public static void yield()** Causes the currently running thread to yield to any other threads of the same priority that are waiting to be scheduled.
2	**public static void sleep(long millisec)** Causes the currently running thread to block for at least the specified number of milliseconds.
3	**public static boolean holdsLock(Object x)** Returns true if the current thread holds the lock on the given Object.
4	**public static Thread currentThread()** Returns a reference to the currently running thread, which is the thread that invokes this method.
5	**public static void dumpStack()** Prints the stack trace for the currently running thread, which is useful when debugging a multithreaded application.

Example

The following ThreadClassDemo program demonstrates some of these methods of the Thread class. Consider a class **DisplayMessage** which implements **Runnable** −

```
// File Name : DisplayMessage.java
// Create a thread to implement Runnable

public class DisplayMessage implements Runnable {
   private String message;

   public DisplayMessage(String message) {
      this.message = message;
   }

   public void run() {
      while(true) {
         System.out.println(message);
      }
   }
}
```

Following is another class which extends the Thread class −

```
// File Name : GuessANumber.java
// Create a thread to extentd Thread

public class GuessANumber extends Thread {
   private int number;
   public GuessANumber(int number) {
      this.number = number;
   }

   public void run() {
      int counter = 0;
      int guess = 0;
      do {
         guess = (int) (Math.random() * 100 + 1);
         System.out.println(this.getName() + " guesses " + guess);
         counter++;
      } while(guess != number);
      System.out.println("** Correct!" + this.getName() + "in" + counter + "guesses.**");
   }
}
```

Following is the main program, which makes use of the above-defined classes −

```
// File Name : ThreadClassDemo.java
public class ThreadClassDemo {

   public static void main(String [] args) {
      Runnable hello = new DisplayMessage("Hello");
      Thread thread1 = new Thread(hello);
      thread1.setDaemon(true);
      thread1.setName("hello");
      System.out.println("Starting hello thread...");
```

```
      thread1.start();

      Runnable bye = new DisplayMessage("Goodbye");
      Thread thread2 = new Thread(bye);
      thread2.setPriority(Thread.MIN_PRIORITY);
      thread2.setDaemon(true);
      System.out.println("Starting goodbye thread...");
      thread2.start();

      System.out.println("Starting thread3...");
      Thread thread3 = new GuessANumber(27);
      thread3.start();
      try {
         thread3.join();
      } catch (InterruptedException e) {
         System.out.println("Thread interrupted.");
      }
      System.out.println("Starting thread4...");
      Thread thread4 = new GuessANumber(75);

      thread4.start();
      System.out.println("main() is ending...");
   }
}
```

This will produce the following result. You can try this example again and again and you will get a different result every time.

Output

```
Starting hello thread...
Starting goodbye thread...
Hello
Hello
Hello
Hello
Hello
Hello
Goodbye
Goodbye
Goodbye
Goodbye
Goodbye
.......
```

Major Java Multithreading Concepts

While doing Multithreading programming in Java, you would need to have the following concepts very handy −

- What is thread synchronization?
- Handling interthread communication
- Handling thread deadlock
- Major thread operations

3. Java - Applet Basics

Previous Page

Next Page

An **applet** is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal.

There are some important differences between an applet and a standalone Java application, including the following −

- An applet is a Java class that extends the java.applet.Applet class.
- A main() method is not invoked on an applet, and an applet class will not define main().
- Applets are designed to be embedded within an HTML page.
- When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine.
- A JVM is required to view an applet. The JVM can be either a plug-in of the Web browser or a separate runtime environment.
- The JVM on the user's machine creates an instance of the applet class and invokes various methods during the applet's lifetime.
- Applets have strict security rules that are enforced by the Web browser. The security of an applet is often referred to as sandbox security, comparing the applet to a child playing in a sandbox with various rules that must be followed.
- Other classes that the applet needs can be downloaded in a single Java Archive (JAR) file.

Life Cycle of an Applet

Four methods in the Applet class gives you the framework on which you build any serious applet −

- **init** − This method is intended for whatever initialization is needed for your applet. It is called after the param tags inside the applet tag have been processed.
- **start** − This method is automatically called after the browser calls the init method. It is also called whenever the user returns to the page containing the applet after having gone off to other pages.
- **stop** − This method is automatically called when the user moves off the page on which the applet sits. It can, therefore, be called repeatedly in the same applet.
- **destroy** − This method is only called when the browser shuts down normally. Because applets are meant to live on an HTML page, you should not normally leave resources behind after a user leaves the page that contains the applet.
- **paint** − Invoked immediately after the start() method, and also any time the applet needs to repaint itself in the browser. The paint() method is actually inherited from the java.awt.

A "Hello, World" Applet

Following is a simple applet named HelloWorldApplet.java −

```
import java.applet.*;
import java.awt.*;

public class HelloWorldApplet extends Applet {
   public void paint (Graphics g) {
      g.drawString ("Hello World", 25, 50);
   }
}
```

These import statements bring the classes into the scope of our applet class −

- java.applet.Applet
- java.awt.Graphics

Without those import statements, the Java compiler would not recognize the classes Applet and Graphics, which the applet class refers to.

The Applet Class

Every applet is an extension of the *java.applet.Applet class*. The base Applet class provides methods that a derived Applet class may call to obtain information and services from the browser context.

These include methods that do the following −

- Get applet parameters
- Get the network location of the HTML file that contains the applet
- Get the network location of the applet class directory
- Print a status message in the browser
- Fetch an image
- Fetch an audio clip
- Play an audio clip
- Resize the applet

Additionally, the Applet class provides an interface by which the viewer or browser obtains information about the applet and controls the applet's execution. The viewer may −

- Request information about the author, version, and copyright of the applet
- Request a description of the parameters the applet recognizes
- Initialize the applet
- Destroy the applet
- Start the applet's execution
- Stop the applet's execution

The Applet class provides default implementations of each of these methods. Those implementations may be overridden as necessary.

The "Hello, World" applet is complete as it stands. The only method overridden is the paint method.

Invoking an Applet

An applet may be invoked by embedding directives in an HTML file and viewing the file through an applet viewer or Java-enabled browser.

The <applet> tag is the basis for embedding an applet in an HTML file. Following is an example that invokes the "Hello, World" applet −

```
<html>
   <title>The Hello, World Applet</title>
   <hr>
   <applet code = "HelloWorldApplet.class" width = "320" height = "120">
      If your browser was Java-enabled, a "Hello, World"
      message would appear here.
   </applet>
   <hr>
</html>
```

Note − You can refer to HTML Applet Tag to understand more about calling applet from HTML.

The code attribute of the <applet> tag is required. It specifies the Applet class to run. Width and height are also required to specify the initial size of the panel in which an applet runs. The applet directive must be closed with an </applet> tag.

If an applet takes parameters, values may be passed for the parameters by adding <param> tags between <applet> and </applet>. The browser ignores text and other tags between the applet tags.

Non-Java-enabled browsers do not process <applet> and </applet>. Therefore, anything that appears between the tags, not related to the applet, is visible in non-Java-enabled browsers.

The viewer or browser looks for the compiled Java code at the location of the document. To specify otherwise, use the codebase attribute of the <applet> tag as shown −

```
<applet codebase = "https://amrood.com/applets" code = "HelloWorldApplet.class"
   width = "320" height = "120">
```

If an applet resides in a package other than the default, the holding package must be specified in the code attribute using the period character (.) to separate package/class components. For example −

```
<applet  = "mypackage.subpackage.TestApplet.class"
   width = "320" height = "120">
```

Getting Applet Parameters

The following example demonstrates how to make an applet respond to setup parameters specified in the document. This applet displays a checkerboard pattern of black and a second color.

The second color and the size of each square may be specified as parameters to the applet within the document.

CheckerApplet gets its parameters in the init() method. It may also get its parameters in the paint() method. However, getting the values and saving the settings once at the start of the applet, instead of at every refresh, is convenient and efficient.

The applet viewer or browser calls the init() method of each applet it runs. The viewer calls init() once, immediately after loading the applet. (Applet.init() is implemented to do nothing.) Override the default implementation to insert custom initialization code.

The Applet.getParameter() method fetches a parameter given the parameter's name (the value of a parameter is always a string). If the value is numeric or other non-character data, the string must be parsed.

The following is a skeleton of CheckerApplet.java −

```
import java.applet.*;
import java.awt.*;

public class CheckerApplet extends Applet {
```

```
   int squareSize = 50;   // initialized to default size
   public void init() {}
   private void parseSquareSize (String param) {}
   private Color parseColor (String param) {}
   public void paint (Graphics g) {}
}
```

Here are CheckerApplet's init() and private parseSquareSize() methods −

```
public void init () {
   String squareSizeParam = getParameter ("squareSize");
   parseSquareSize (squareSizeParam);

   String colorParam = getParameter ("color");
   Color fg = parseColor (colorParam);

   setBackground (Color.black);
   setForeground (fg);
}

private void parseSquareSize (String param) {
   if (param == null) return;
   try {
      squareSize = Integer.parseInt (param);
   } catch (Exception e) {
      // Let default value remain
   }
}
```

The applet calls parseSquareSize() to parse the squareSize parameter. parseSquareSize() calls the library method Integer.parseInt(), which parses a string and returns an integer. Integer.parseInt() throws an exception whenever its argument is invalid.

Therefore, parseSquareSize() catches exceptions, rather than allowing the applet to fail on bad input.

The applet calls parseColor() to parse the color parameter into a Color value. parseColor() does a series of string comparisons to match the parameter value to the name of a predefined color. You need to implement these methods to make this applet work.

Specifying Applet Parameters

The following is an example of an HTML file with a CheckerApplet embedded in it. The HTML file specifies both parameters to the applet by means of the <param> tag.

```
<html>
   <title>Checkerboard Applet</title>
   <hr>
   <applet code = "CheckerApplet.class" width = "480" height = "320">
      <param name = "color" value = "blue">
      <param name = "squaresize" value = "30">
   </applet>
   <hr>
</html>
```

Note − Parameter names are not case sensitive.

Application Conversion to Applets

It is easy to convert a graphical Java application (that is, an application that uses the AWT and that you can start with the Java program launcher) into an applet that you can embed in a web page.

Following are the specific steps for converting an application to an applet.

- Make an HTML page with the appropriate tag to load the applet code.
- Supply a subclass of the JApplet class. Make this class public. Otherwise, the applet cannot be loaded.
- Eliminate the main method in the application. Do not construct a frame window for the application. Your application will be displayed inside the browser.
- Move any initialization code from the frame window constructor to the init method of the applet. You don't need to explicitly construct the applet object. The browser instantiates it for you and calls the init method.
- Remove the call to setSize; for applets, sizing is done with the width and height parameters in the HTML file.
- Remove the call to setDefaultCloseOperation. An applet cannot be closed; it terminates when the browser exits.
- If the application calls setTitle, eliminate the call to the method. Applets cannot have title bars. (You can, of course, title the web page itself, using the HTML title tag.)
- Don't call setVisible(true). The applet is displayed automatically.

Event Handling

Applets inherit a group of event-handling methods from the Container class. The Container class defines several methods, such as processKeyEvent and processMouseEvent, for handling particular types of events, and then one catch-all method called processEvent.

In order to react to an event, an applet must override the appropriate event-specific method.

```
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.applet.Applet;
import java.awt.Graphics;

public class ExampleEventHandling extends Applet implements MouseListener {
  StringBuffer strBuffer;

  public void init() {
    addMouseListener(this);
    strBuffer = new StringBuffer();
    addItem("initializing the apple ");
  }

  public void start() {
    addItem("starting the applet ");
  }

  public void stop() {
    addItem("stopping the applet ");
  }
```

```
  public void destroy() {
    addItem("unloading the applet");
  }

  void addItem(String word) {
    System.out.println(word);
    strBuffer.append(word);
    repaint();
  }

  public void paint(Graphics g) {
    // Draw a Rectangle around the applet's display area.
    g.drawRect(0, 0,
    getWidth() - 1,
    getHeight() - 1);

    // display the string inside the rectangle.
    g.drawString(strBuffer.toString(), 10, 20);
  }


  public void mouseEntered(MouseEvent event) {
  }
  public void mouseExited(MouseEvent event) {
  }
  public void mousePressed(MouseEvent event) {
  }
  public void mouseReleased(MouseEvent event) {
  }
  public void mouseClicked(MouseEvent event) {
    addItem("mouse clicked! ");
  }
}
```

Now, let us call this applet as follows −

```
<html>
  <title>Event Handling</title>
  <hr>
  <applet code = "ExampleEventHandling.class"
    width = "300" height = "300">
  </applet>
  <hr>
</html>
```

Initially, the applet will display "initializing the applet. Starting the applet." Then once you click inside the rectangle, "mouse clicked" will be displayed as well.

Displaying Images

An applet can display images of the format GIF, JPEG, BMP, and others. To display an image within the applet, you use the drawImage() method found in the java.awt.Graphics class.

Following is an example illustrating all the steps to show images −

```
import java.applet.*;
import java.awt.*;
import java.net.*;

public class ImageDemo extends Applet {
   private Image image;
   private AppletContext context;

   public void init() {
      context = this.getAppletContext();
      String imageURL = this.getParameter("image");
      if(imageURL == null) {
         imageURL = "java.jpg";
      }
      try {
         URL url = new URL(this.getDocumentBase(), imageURL);
         image = context.getImage(url);
      } catch (MalformedURLException e) {
         e.printStackTrace();
         // Display in browser status bar
         context.showStatus("Could not load image!");
      }
   }

   public void paint(Graphics g) {
      context.showStatus("Displaying image");
      g.drawImage(image, 0, 0, 200, 84, null);
      g.drawString("www.javalicense.com", 35, 100);
   }
}
```

Now, let us call this applet as follows −

```
<html>
   <title>The ImageDemo applet</title>
   <hr>
   <applet code = "ImageDemo.class" width = "300" height = "200">
      <param name = "image" value = "java.jpg">
   </applet>
   <hr>
</html>
```

Playing Audio

An applet can play an audio file represented by the AudioClip interface in the java.applet package. The AudioClip interface has three methods, including −

- **public void play()** − Plays the audio clip one time, from the beginning.
- **public void loop()** − Causes the audio clip to replay continually.
- **public void stop()** − Stops playing the audio clip.

To obtain an AudioClip object, you must invoke the getAudioClip() method of the Applet class. The getAudioClip() method returns immediately, whether or not the URL resolves to an actual audio file. The audio file is not downloaded until an attempt is made to play the audio clip.

Following is an example illustrating all the steps to play an audio −

```
import java.applet.*;
import java.awt.*;
import java.net.*;

public class AudioDemo extends Applet {
   private AudioClip clip;
   private AppletContext context;

   public void init() {
      context = this.getAppletContext();
      String audioURL = this.getParameter("audio");
      if(audioURL == null) {
         audioURL = "default.au";
      }
      try {
         URL url = new URL(this.getDocumentBase(), audioURL);
         clip = context.getAudioClip(url);
      } catch (MalformedURLException e) {
         e.printStackTrace();
         context.showStatus("Could not load audio file!");
      }
   }

   public void start() {
      if(clip != null) {
         clip.loop();
      }
   }

   public void stop() {
      if(clip != null) {
         clip.stop();
      }
   }
}
```

Now, let us call this applet as follows −

```
<html>
   <title>The ImageDemo applet</title>
   <hr>
   <applet code = "ImageDemo.class" width = "0" height = "0">
      <param name = "audio" value = "test.wav">
   </applet>
   <hr>
</html>
```

You can use test.wav on your PC to test the above example.

4. Top 100 Java Interview Questions and Answers (Download PDF)

Download PDF

We have compiled the most frequently asked Java Interview Questions and Answers that will help you prepare for the Basic Java interview questions that an interviewer might ask you during your interview. In this list of Basic Java interview questions, we have covered all commonly asked basic and advanced Core Java interview questions with detailed answers to help you clear the job interview.

The following list contains 100 important Core Java interview questions for freshers as well as Java interview questions and answers for experienced programmers to help them prepare for the interview. This detailed guide of interview questions for Java Programming will help you to crack your Job interview easily.

Core Java Interview Questions and Answers for Freshers and Experienced

Q1. What is the difference between an Inner Class and a Sub-Class?

Ans: 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.

Q2. What are the various access specifiers for Java classes?

Ans: 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.

Q3. What's the purpose of Static methods and static variables?

Ans: 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.

Q4. What is data encapsulation and what's its significance?

Ans: 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.

Q5. 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.

Q6. What are Loops in Java? What are three types of loops?

Ans: 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.

Q7: What is an infinite Loop? How infinite loop is declared?

Ans: 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
}
```

Q8. What is the difference between continue and break statement?

Ans: 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");
}
```

Q9. What is the difference between double and float variables in Java?

Ans: 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.

Q10. What is Final Keyword in Java? Give an example.

Ans: 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.

Q11. What is ternary operator? Give an example.

Ans: 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);
    }
}
```

Q12: How can you generate random numbers in Java?

Ans:

- 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

Q13. What is default switch case? Give example.

Ans: 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 {
```

```
   int score = 4;
   public static void main(String args[]) {
      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");
      }
   }
}
```

Q14. What's the base class in Java from which all classes are derived?

Ans: java.lang.object

Q15. Can main() method in Java can return any data?

Ans: In java, main() method can't return any data and hence, it's always declared with a void return type.

Q16. What are Java Packages? What's the significance of packages?

Ans: 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.

Q17. Can we declare a class as Abstract without having any abstract method?

Ans: 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.

Q18. What's the difference between an Abstract Class and Interface in Java?

Ans: 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.

Q19. What are the performance implications of Interfaces over abstract classes?

Ans: 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.

Q20. Does Importing a package imports its sub-packages as well in Java?

Ans: 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.*

Q21. Can we declare the main method of our class as private?

Ans: 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.

Q22. How can we pass argument to a function by reference instead of pass by value?

Ans: In java, we can pass argument to a function only by value and not by reference.

Q23. How an object is serialized in java?

Ans: 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.

Q24. When we should use serialization?

Ans: 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.

Q25. Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling?

Ans: 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.

Q26. Is there any way to skip Finally block of exception even if some exception occurs in the exception block?

Ans: 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);
```

Q27. When the constructor of a class is invoked?

Ans: 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();
   }
}
```

Q28. Can a class have multiple constructors?

Ans: 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.

Q29. Can we override static methods of a class?

Ans: 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.

Q30. 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

Displaying from superclass

Q31. Is String a data type in java?

Ans: 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.

Q32. 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";
```

Ans: In the above example, two objects of Java.Lang.String class are created. s1 and s3 are references to same object.

Q33. Why Strings in Java are called as Immutable?

Ans: 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";
```

Q34. What's the difference between an array and Vector?

Ans: 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.

Q35. What is multi-threading?

Ans: 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.

Q36. Why Runnable Interface is used in Java?

Ans: Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading.

Q37. What are the two ways of implementing multi-threading in Java?

Ans: 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.

Q38. When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer?

Ans: 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.

Q39. What's the purpose of using Break in each case of Switch Statement?

Ans: 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.

Q40. How garbage collection is done in Java?

Ans: 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.

Q41. How we can execute any code even before main method?

Ans: 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.

Q42. Can a class be a super class and a sub-class at the same time? Give example.

Ans: 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 {

.....

}
```

Q43. How objects of a class are created if no constructor is defined in the class?

Ans: 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.

Q44. In multi-threading how can we ensure that a resource isn't used by multiple threads simultaneously?

Ans: 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.

Q45. Can we call the constructor of a class more than once for an object?

Ans: 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.

Q46. 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?

Ans: 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.

Q47. Can we have two methods in a class with the same name?

Ans: 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");

    }

}
```

Q48. How can we make copy of a java object?

Ans: 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.

Q49. What's the benefit of using inheritance?

Ans: 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.

Q50. What's the default access specifier for variables and methods of a class?

Ans: 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.

Q51. Give an example of use of Pointers in Java class.

Ans: There are no pointers in Java. So we can't use concept of pointers in Java.

Q52. How can we restrict inheritance for a class so that no class can be inherited from it?

Ans: 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
}
```

Q53. What's the access scope of Protected Access specifier?

Ans: 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	Worl
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Q54. What's difference between Stack and Queue?

Ans: 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.

Q55. In java, how we can disallow serialization of variables?

Ans: 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
}
```

Q56. How can we use primitive data types as objects?

Ans: 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.

Q57. Which types of exceptions are caught at compile time?

Ans: 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.

Q58. Describe different states of a thread.

Ans: 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.

Q59. Can we use a default constructor of a class even if an explicit constructor is defined?

Ans: 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.

Q60. Can we override a method by using same method name and arguments but different return types?

Ans: 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.

Q61.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++);
            }
}
```

Ans: In this case postfix ++ operator is used which first returns the value and then increments. Hence it's output will be 4.

Q61. A person says that he compiled a java class successfully without even having a main method in it? Is it possible?

Ans: 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.

Q62. Can we call a non-static method from inside a static method?

Ans: 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.

Q63. What are the two environment variables that must be set in order to run any Java programs?

Ans: Java programs can be executed in a machine only once following two environment variables have been properly set:

1. PATH variable
2. CLASSPATH variable

Q64. Can variables be used in Java without initialization?

Ans: 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.

Q65. Can a class in Java be inherited from more than one class?

Ans: In Java, a class can be derived from only one class and not from multiple classes. Multiple inheritances is not supported by Java.

Q66. Can a constructor have different name than a Class name in Java?

Ans: 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.

Q67. What will be the output of Round(3.7) and Ceil(3.7)?

Ans: Round(3.7) returns 4 and Ceil(3.7) returns 4.

Q68: Can we use goto in Java to go to a particular line?

Ans: In Java, there is not goto keyword and java doesn't support this feature of going to a particular labeled line.

Q69. Can a dead thread be started again?

Ans: In java, a thread which is in dead state can't be started again. There is no way to restart a dead thread.

Q70. Is the following class declaration correct?

Ans:

```
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.

Q71. Is JDK required on each machine to run a Java program?

Ans: 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.

Q72. What's the difference between comparison done by equals method and == operator?

Ans: 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");

    }

  }

}
```

Q73. Is it possible to define a method in Java class but provide it's implementation in the code of another language like C?

Ans: 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.

Q74. How are destructors defined in Java?

Ans: 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.

Q75. Can a variable be local and static at the same time?

Ans: No a variable can't be static as well as local at the same time. Defining a local variable as static gives compilation error.

Q76. Can we have static methods in an Interface?

Ans: 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.

Q77. In a class implementing an interface, can we change the value of any variable defined in the interface?

Ans: 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.

Q78. Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory?

Ans: 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.

Q79. Can we have any other return type than void for main method?

Ans: 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)

Q80. I want to re-reach and use an object once it has been garbage collected. How it's possible?

Ans: 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.

Q81. In Java thread programming, which method is a must implementation for all threads?

Ans: Run() is a method of Runnable interface that must be implemented by all threads.

Q82. 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.

Q83. How can an exception be thrown manually by a programmer?

Ans: 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 ");
        }
    }
```

Q84. 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?

Ans: 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.

Q85. How objects are stored in Java?

Ans: 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.

Q86. How can we find the actual size of an object on the heap?

Ans: In java, there is no way to find out the exact size of an object on the heap.

Q87. 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

Ans: 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.

Q88. What happens if an exception is not handled in a program?

Ans: 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.

Q89. I have multiple constructors defined in a class. Is it possible to call a constructor from another constructor's body?

Ans: If a class has multiple constructors, it's possible to call one constructor from the body of another one using **this()**.

Q90. What's meant by anonymous class?

Ans: 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;

      }

   }
```

Q91. Is there a way to increase the size of an array after its declaration?

Ans: 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.

Q92. If an application has multiple classes in it, is it okay to have a main method in more than one class?

Ans: 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.

Q93. I want to persist data of objects for later use. What's the best approach to do so?

Ans: The best way to persist data for future use is to use the concept of serialization.

Q94. What is a Local class in Java?

Ans: 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.

Q95. String and StringBuffer both represent String objects. Can we compare String and StringBuffer in Java?

Ans: 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.

Q96. Which API is provided by Java for operations on set of objects?

Ans: 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.

Q97. Can we cast any other type to Boolean Type with type casting?

Ans: 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.

Q98. Can we use different return types for methods when overridden?

Ans: 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

    }

}
```

Q99. What's the base class of all exception classes?

Ans: In Java, **Java.lang.Throwable** is the super class of all exception classes and all exception classes are derived from this base class.

Q100. What's the order of call of constructors in inheritance?

Ans: 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 interview questions pdf will help you to clear the doubts about Java interview questions and will also help you to crack the interview.