1) What is Java?

[Java](https://www.javatpoint.com/java-tutorial) is the high-level, [object-oriented](https://www.javatpoint.com/java-oops-concepts), robust, secure programming language, platform-independent, high performance, Multithreaded, and portable programming language. It was developed by [**James Gosling**](https://www.javatpoint.com/james-gosling-father-of-java) in June 1991. It can also be known as the platform as it provides its own JRE and API.

### 3) List the features of Java Programming language.

There are the following features in Java Programming Language.

* **Simple:** Java is easy to learn. The syntax of Java is based on C++ which makes easier to write the program in it
* **Object-Oriented:** Java follows the object-oriented paradigm which allows us to maintain our code as the combination of different type of objects that incorporates both data and behavior
* **Portable:** Java supports read-once-write-anywhere approach. We can execute the Java program on every machine. Java program (.java) is converted to bytecode (.class) which can be easily run on every machine
* **Platform Independent:** Java is a platform independent programming language. It is different from other programming languages like C and C++ which needs a platform to be executed. Java comes with its platform on which its code is executed. Java doesn't depend upon the operating system to be executed
* **Secured:** Java is secured because it doesn't use explicit pointers. Java also provides the concept of ByteCode and Exception handling which makes it more secured
* **Robust:** Java is a strong programming language as it uses strong memory management. The concepts like Automatic garbage collection, Exception handling, etc. make it more robust
* **Architecture Neutral:** Java is architectural neutral as it is not dependent on the architecture. In C, the size of data types may vary according to the architecture (32 bit or 64 bit) which doesn't exist in Java
* **Interpreted:** Java uses the Just-in-time (JIT) interpreter along with the compiler for the program execution.
* **High Performance:** Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++)
* **Multithreaded:** We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.
* **Distributed:** Java is distributed because it facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machi

### 4) What do you understand by Java virtual machine?

[Java Virtual Machine](https://www.javatpoint.com/jvm-java-virtual-machine) is a virtual machine that enables the computer to run the Java program. JVM acts like a run-time engine which calls the main method present in the Java code. JVM is the specification which must be implemented in the computer system. The Java code is compiled by JVM to be a Bytecode which is machine independent and close to the native code. JVM, loads, verifies and executes Java bytecode. It is known as the interpreter or the core of Java programming language because it executes Java programming. JVM is apart of JRE

6) How many types of memory areas are allocated by JVM?

JVM Acgitecture?

Many types:

**1.Class Loader**: Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program, it is loaded first by the classloader

2.**Class(Method) Area:** Class Area stores per-class structures such as the runtime constant pool, field, method data, and the code for methods.

1. **Heap:** It is the runtime data area in which the memory is allocated to the objects
2. **Stack:** Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return. Each thread has a private JVM stack, created at the same time as the thread. A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes. **Stack space is mainly used for storing order of method execution and local variables**
3. **Program Counter Register:** PC (program counter) register contains the address of the Java virtual machine instruction currently being executed.
4. **Native Method Stack:** It contains all the native methods used in the application.
5. **Execution Engine** : It contains virtual processor, Just In compiler, Interpreter
6. **Java Native Interface :j** ava Native Interface (JNI) is a framework which provides an interface to communicate with another application written in another language like C, C++, Assembly etc. Java uses JNI framework to send output to the Console or interact with OS libraries.
7. **Q. Just-In-Time(JIT) compiler:** It is used to improve the performance. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here, the term "compiler" refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

7.Is Java 100% Object Oriented Language? Java is not 100 percent object oriented programming because it uses primitive data type (int float long)

Q 8. What is JRE, and why is it required? : JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime

Q 9. What is JDK, and why is it required? : JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and [applets](https://www.javatpoint.com/java-applet). It physically exists. It contains JRE + development tools. The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application

Q.What is Data type in java? And its Data type?

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

1. **Primitive data types:** The primitive data types include boolean, char, byte, short, int, long, float and double.
2. **Non-primitive data types:** The non-primitive data types include [Classes](https://www.javatpoint.com/object-and-class-in-java), [Interfaces](https://www.javatpoint.com/interface-in-java), and [Arrays](https://www.javatpoint.com/array-in-java).

Q. What is the variable in java? : A variable is the name of a reserved area allocated in memoryA variable is container placeholder which holds value which java program executed A variable assigned with the data type Variable is a name of memory location its value can be changed we can change the value of variable anyways

1. Q Type Of Variable?
2. 1)Local Variable : A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists A local variable cannot be defined with "static" keyword.
3. 2)Instance Variable : A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as [static](https://www.javatpoint.com/static-keyword-in-java). Instance variable and instance member function belongs to object not class We can acess instance variable in method of same class.
4. 3)Static Variable : A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory. If we want call static variable and method the we can call without creating object . We can call static variable and static method by classname.variable name

### Q.What are the various access specifiers in Java?

In Java, access specifiers are the keywords which are used to define the access scope of the method, class, or a variable. In Java, there are four access specifiers given below.

* **Public** The classes, methods, or variables which are defined as public, can be accessed by any class or method.
* **Protected** Protected can be accessed by the class of the same package, or by the sub-class of this class, or within the same class.
* **Default** Default are accessible within the package only. By default, all the classes, methods, and variables are of default scope.
* **Private** The private class, methods, or variables defined as private can be accessed within the class only.

Q.11.What is a Method in Java? : In Java, a method is like a function which is used to expose the behavior of an object.

### Predefined Method

In Java, predefined methods are the method that is already defined in the Java class libraries is known as predefined methods.

### User-defined Method

The method written by the user or programmer is known as **a user-defined** method. These methods are modified according to the requirement.

Q.What is finalized method in java? : Finalize() is the method of Object class. This method is called just before an object is garbage collected. finalize() method overrides to dispose system resources, perform clean-up activities and minimize memory leaks. This used in garbage collection concept

Q 12 What is new keyword in java? The new keyword is used to allocate memory at runtime. All objects get memory in Heap memory area.

Q.13 What is use of Final Keyword? Final is the keyword which is used as variable and method and classname . If you make any variable as final, you cannot change the value of final variable(It will be constant). If you make any method as final, you cannot override it. If you make any class as final, you cannot extend it.

### Q) Is final method inherited?: Yes, final method is inherited but you cannot override it.

### 18) What is the purpose of static methods and variables?

The methods or variables defined as static are shared among all the objects of the class. The static is the part of the class and not of the object. The static variables are stored in the class area, and we do not need to create the object to access such variables. Therefore, static is used in the case, where we need to define variables or methods which are common to all the objects of the class.

For example, In the class simulating the collection of the students in a college, the name of the college is the common attribute to all the students. Therefore, the college name will be defined as **static**

**Q.Why is the main method static?**

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

Q. **Can we override the static methods?**

No, we can't override static methods

### Q.What is the static block?

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

Q **What if the static modifier is removed from the signature of the main method?**

Program compiles. However, at runtime, It throws an error "NoSuchMethodError.

# Interview Question on **Java Control Statements**

D,

### Decision-Making statements: Decision-making statements evaluate the Boolean expression and control the program flow depending upon the result of the condition provided. There are two types of decision-making statements in Java, i.e., If statement and switch statement.

### a) If Statement:

In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false. In Java, there are four types of if-statements given below

1. Simple if statement
2. if-else statement
3. if-else-if ladder
4. Nested if-statement

### 1) Simple if statement:

It is the most basic statement among all control flow statements in Java. It evaluates a Boolean expression and enables the program to enter a block of code if the expression evaluates to true.

### 2) if-else statement

The [if-else statement](https://www.javatpoint.com/java-if-else) is an extension to the if-statement, which uses another block of code, i.e., else block. The else block is executed if the condition of the if-block is evaluated as false.

### 3) if-else-if ladder:

The if-else-if statement contains the if-statement followed by multiple else-if statements. In other words, we can say that it is the chain of if-else statements that create a decision tree where the program may enter in the block of code where the condition is true. We can also define an else statement at the end of the chain.

### 4. Nested if-statement

In nested if-statements, the if statement can contain a **if** or **if-else** statement inside another if or else-if statement.

### Switch Statement:

In Java, [Switch statements](https://www.javatpoint.com/java-switch) are similar to if-else-if statements. The switch statement contains multiple blocks of code called cases and a single case is executed based on the variable which is being switched. The switch statement is easier to use instead of if-else-if statements. It also enhances the readability of the program.

* The case variables can be int, short, byte, char, or enumeration. String type is also supported since version 7 of Java
* Cases cannot be duplicate
* Default statement is executed when any of the case doesn't match the value of expression. It is optional.
* Break statement terminates the switch block when the condition is satisfied.  
  It is optional, if not used, next case is executed.
* While using switch statements, we must notice that the case expression will be of the same type as the variable. However, it will also be a constant value.

Loop Statements

In programming, sometimes we need to execute the block of code repeatedly while some condition evaluates to true. However, loop statements are used to execute the set of instructions in a repeated order. The execution of the set of instructions depends upon a particular condition.

In Java, we have three types of loops that execute similarly. However, there are differences in their syntax and condition checking time.

1. for loop
2. while loop
3. do-while lo

### Java for loop

In Java, [for loop](https://www.javatpoint.com/java-for-loop) is similar to [C](https://www.javatpoint.com/c-programming-language-tutorial) and [C++](https://www.javatpoint.com/cpp-tutorial). It enables us to initialize the loop variable, check the condition, and increment/decrement in a single line of code. We use the for loop only when we exactly know the number of times, we want to execute the block of code.

### Java while loop

The [while loop](https://www.javatpoint.com/java-while-loop) is also used to iterate over the number of statements multiple times. However, if we don't know the number of iterations in advance, it is recommended to use a while loop. Unlike for loop, the initialization and increment/decrement doesn't take place inside the loop statement in while loop.

It is also known as the entry-controlled loop since the condition is checked at the start of the loop. If the condition is true, then the loop body will be executed; otherwise, the statements after the loop will be executed

### Java do-while loop

The [do-while loop](https://www.javatpoint.com/java-do-while-loop) checks the condition at the end of the loop after executing the loop statements. When the number of iteration is not known and we have to execute the loop at least once, we can use do-while loop.

It is also known as the exit-controlled loop since the condition is not checked in advance.

### Jump Statements

Jump statements are used to transfer the control of the program to the specific statements. In other words, jump statements transfer the execution control to the other part of the program. There are two types of jump statements in Java, i.e., break and continue.

### Java break statement

As the name suggests, the [break statement](https://www.javatpoint.com/java-break) is used to break the current flow of the program and transfer the control to the next statement outside a loop or switch statement. However, it breaks only the inner loop in the case of the nested loop.

The break statement cannot be used independently in the Java program, i.e., it can only be written inside the loop or switch statement.

### Java continue statement

Unlike break statement, the [continue statement](https://www.javatpoint.com/java-continue) doesn't break the loop, whereas, it skips the specific part of the loop and jumps to the next iteration of the loop immediately.

Interview question on Classes and Object

Q 10 .What is an Object in Java? An object is an instance of a class . Object is real workd entity which /IP

Thas its state and behaviour. An object is a runtime entity. An object in Java is the physical as well as a logical entity,

* **State:** represents the data (value) of an object.
* **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
* **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

Q.11.What is a Class in Java? : A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical. A class contain fields, methods and constructor and blocks

**Question onOOPS:**

Q why the java is called object programming language?

Java Language is considered an object-oriented language because it is based on the concept of objects and classes. Without the creation of objects and classes, it is ‘/ c impossible to write any code in Java. Java supports the concepts of OOPS - Inheritance, Data abstraction, polymorphism, and data encapsulation.

Q.Why the java is not pure object programming language?

Java is not a pure object oriented language because it supports Primitive datatype such as int, byte, long? etc, to be used, which are not objects.  
Java provides wrapper class for int, long, etc? But still int, long, float,etc? are not classes. Integer, Float, Long only classes. It also used static In Java, a class declared as static can be used without the use of an object. We cannot call that function or variable using a dot(.) if we are using a static variable or a static class.

Q.what is class in java?

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical. Inside a class, we define variables, constants, member functions, and other functionality. It does not consume memory at run time. It is user defined data type

* Q what is object? An object is a real-world entity that has attributes, behavior, and properties. It is referred to as an instance of the class. It contains member functions, variables that we have defined in the class. It occupies space in the memory. Different objects have different states or attributes, and behaviors. An object is *a runtime entity*. We can cereate object using new keyword

Q 16.What is Polymorphism? Polymorphism means we can perform single action in different ways. polymorphism refers to the ability of a class to provide different implementations of a method, depending on the type of object that is passed to the method Polymorphism is **the method in an object-oriented programming language that performs different things as per the object's class, which calls it**. With Polymorphism, a message is sent to multiple class objects, and every object responds appropriately according to the properties of the class In java Polymorphism done by method overloading and method overriding

### Q) Why Method Overloading is not possible by changing the return type of method only? : In java, method overloading is not possible by changing the return type of the method only because of ambiguity

### Q Can we overload java main() method?

### Yes, by method overloading. You can have any number of main methods in a class by method overloading. But [JVM](https://www.javatpoint.com/jvm-java-virtual-machine) calls main() method which receives string array as arguments only.

### Q What is upcasting? When an overridden method of child class is called through its parent type reference in Java, it is called upcasting. In this process, the object type represents the method or functionality that will be invoked. This decision is made during the runtime, and hence, the name run time polymorphism is given to the process.

Q.What is Abstraction in Java? abstraction is the process of hiding certain details and showing only essential information to the user. Abstraction can be achieved with either abstract classes or interfaces Abstraction is **a process of hiding the implementation details and showing only functionality to the user**.

Q. What is Abstract Class in Java? : A class which is declared with the abstract keyword is known as an abstract class in [Java](https://www.javatpoint.com/java-tutorial). It can have abstract and non-abstract methods (method with the body). Abstract class contain abstract method and non abstract method and static method also It needs to be extended and its method implemented. It cannot be instantiated(We can not create its object ). An abstract class can have a data member, abstract method, method body (non-abstract method), constructor, and even main() method.

* An abstract class must be declared with an abstract keyword.
* It can have abstract and non-abstract methods.
* It cannot be instantiated.
* It can have [constructors](https://www.javatpoint.com/java-constructor) and static methods also.
* It can have final methods which will force the subclass not to change the body of the method.

#### **Rule: If there is an abstract method in a class, that class must be abstract.**

Q.What is Abstract Method? A method which is declared as a abstract and does not have its implementation is abstract method Its implementation is provided by creating the another classs which is child class. Abstract method have method signature and parameter dosnt have body

Q. What is Interface in java? An **interface in Java** is a blueprint of a class. It has static constants and abstract methods. The interface in Java is a mechanism to achieve [*abstraction*](https://www.javatpoint.com/abstract-class-in-java). There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple [inheritance in Java](https://www.javatpoint.com/inheritance-in-java) In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body. It cannot be instantiated just like the abstract class. We can not create object of interface

## Q.**Why use Java interface?**

* It is used to achieve abstraction.
* By interface, we can support the functionality of multiple inheritance.
* It can be used to achieve loose coupling.

## **Q) Multiple inheritance is not supported through class in java, but it is possible by an interface, why?**

 multiple inheritance is not supported in the case of [class](https://www.javatpoint.com/object-and-class-in-java) because of ambiguity. **whenever a sub or child class wants to access the property of two or more parent or super class that have same method, java compiler can't decide which class method it should be inherit**.

However, it is supported in case of an interface because there is no ambiguity. It is because its implementation is provided by the implementation class.   an interface is defined by someone else, but its implementation is provided by different implementation providers.

**10. What is Concrete method in Java?**

Ans: A concrete method in Java is a method which has always the body. It is also called a complete method in java.

Q. When u create an class which is parent class of that class?

Ans: When u create an class by default object will be the parent class of that class

Can You Override Private Method in Java ?  
  
No, We can not override the private method in Java, just like [we can not override the static method in Java](http://java67.blogspot.sg/2012/08/can-we-override-static-method-in-java.html). Like static methods, the private method in Java is also bonded during compile time using static binding by Type information and doesn't depend on what kind of object a particular reference variable is holding. Since [method overriding](http://java67.blogspot.sg/2012/08/what-is-method-overriding-in-java-example-tutorial.html) works on [dynamic binding](http://javarevisited.blogspot.com/2012/03/what-is-static-and-dynamic-binding-in.html), it's not possible to override the private method in Java.

private methods are not even visible to the Child class, they are only visible and accessible in the class on which they are declared. private keyword provides the highest level of

Q.What is Encapsulation in java?

**Encapsulation in Java** is a process of wrapping code and data together into a single unit, for example, a capsule which is mixed of several medicines. We can create a fully encapsulated class in Java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.

Q.Why we are required abstract class class to achieve abstraction weather we have interface?

OR Q.Why we are required interface to achieve abstraction weather we have abstract class?

Ans An abstract class allows you to create functionality that subclasses can implement or override. An interface only allows you to define functionality, not implement it. And whereas a class can extend only one abstract class, it can take advantage of multiple interfaces.

Q Difference between Throws and throw keyword?

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. no.** | **Basis of Differences** | **Throw** | **Throws** |
| 1. | Definition | Java throw keyword is used throw an exception explicitly in the code, inside the function or the block of code. | Java throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code. |
| 2. | Type of exception Using throw keyword, we can only propagate unchecked exception i.e., the checked exception cannot be propagated using throw only. | Using throws keyword, we can declare both checked and unchecked exceptions. However, the throws keyword can be used to propagate checked exceptions only. |  |
| 3. | Syntax | The throw keyword is followed by an instance of Exception to be thrown. | The throws keyword is followed by class names of Exceptions to be thrown. |
| 4. | Declaration | throw is used within the method. | throws is used with the method signature. |
| 5. | Internal implementation | We are allowed to throw only one exception at a time i.e. we cannot throw multiple exceptions. | We can declare multiple exceptions using throws keyword that can be thrown by the method. For example, main() throws IOException, SQLException. |

Q.What is exception in java?

In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime. Exception is an abnormal condition , unwanted ,unexpected event which occur during the execution of program When an exception occur within a method it create object This object is called as exception object. It contains information and exception such as name and description of the exception

Q.What is exception handling?

The **Exception Handling in Java** is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained. Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc. The core advantage of exception handling is **to maintain the normal flow of the application**

Q.Types Of Exception?

1)Checked Exception Or CompileTime Exception

Q.What is Super Keyword in java? When we use super keyword? Why we use Super keyword in java?

Superkeyword in java is reference variable for immediate parent class object. So whenever we create of instant of subclass (Childclass) the object of parent class implicitely call by its reference variable using super keyword when we have access the instance member variable of parent class then we used super keyword . It is used when parent class and child class have same instant variable same

Q.What is This keyword in java?When we use this keyword ?Why we use This Keyword?

There can be a lot of usage of **Java this keyword**. In Java, this is a **reference variable** that refers to the current object. If the instance member variable and local variable are same so to differentiate we can use this keyword

1. [this can be used to refer current class instance variable.](https://www.javatpoint.com/this1)
2. [this can be used to invoke current class method (implicitly)](https://www.javatpoint.com/this2)

Q. What is Collection in java?

Java collection is single unit of object The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects. It give the facility to maintain multiple object in a single unit

#### Q **What is Collection framework**

The Collection framework represents a unified architecture for storing and manipulating a group of objects. It has:

1. Interfaces and its implementations, i.e., classes
2. Algorithm

Q.Expalin Hierarchy of Collection Framework?