## 1) What is a method in Java?

Ans:A **method** is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation. It is used to achieve the **reusability** of code. We write a method once and use it many times. We do not require to write code again and again. It also provides the **easy modification** and **readability** of code, just by adding or removing a chunk of code. The method is executed only when we call or invoke it.The most important method in Java is the **main()** method. If you want to read more about the main() method

2)what is mean by abstract keyword?

Ans: The abstract keyword is used to achieve abstraction in Java. It is a non-access modifier which is used to create abstract class and method.

The role of an abstract class is to contain abstract methods. However, it may also contain non-abstract methods. The method which is declared with abstract keyword and doesn't have any implementation is known as an abstract method.

3) Abstract Classes and Methods?

Ans: **Abstract class: is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).** Abstract method: can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from).

4)How to implement abstract method?

Ans:

5)What is mean by array in java?

Ans: **Java array** is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array

6)What are the types of array?

### **Ans:Types of Array in java**

There are two types of array.

* Single Dimensional Array
* Multidimensional Array

7)Which method is used to represent the size of array?

Ans: array.length

8)What is string in java & in which package string is present?

Ans: string is an object that represents a sequence of characters. The java.lang.String class is used to create a string object.

### **How to create a string object?**

There are two ways to create String object:

1. By string literal
2. By new keyword

9)Why string is immutable?

Ans: The String is immutable in Java **because of the security, synchronization and concurrency, caching, and class loading**. The reason of making string final is to destroy the immutability and to not allow others to extend it. The String objects are cached in the String pool, and it makes the String immutable.

10)In how many ways we can create the string object?

There are two ways to create String object:

1. By string literal
2. By new keyword

11)What are the methods present in string class?

Ans:

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Return Type** |
| [charAt()](https://www.w3schools.com/java/ref_string_charat.asp) | Returns the character at the specified index (position) | char |
| [codePointAt()](https://www.w3schools.com/java/ref_string_codepointat.asp) | Returns the Unicode of the character at the specified index | int |
| [codePointBefore()](https://www.w3schools.com/java/ref_string_codepointbefore.asp) | Returns the Unicode of the character before the specified  index | int |
| [codePointCount()](https://www.w3schools.com/java/ref_string_codepointcount.asp) | Returns the number of Unicode values found in a string. | int |
| [compareTo()](https://www.w3schools.com/java/ref_string_compareto.asp) | Compares two strings lexicographically | int |
| [compareToIgnoreCase()](https://www.w3schools.com/java/ref_string_comparetoignorecase.asp) | Compares two strings lexicographically, ignoring case  differences | int |
|  |  |  |
|  |  |  |
| [concat()](https://www.w3schools.com/java/ref_string_concat.asp) | Appends a string to the end of another string | String |
| [contains()](https://www.w3schools.com/java/ref_string_contains.asp) | Checks whether a string contains a sequence of characters | boolean |
| [contentEquals()](https://www.w3schools.com/java/ref_string_contentequals.asp) | Checks whether a string contains the exact same sequence  of characters of the specified CharSequence or StringBuffer | boolean |
|  |  |  |
| [copyValueOf()](https://www.w3schools.com/java/ref_string_copyvalueof.asp) | Returns a String that represents the characters of the  character array | String |
|  |  |  |
| [endsWith()](https://www.w3schools.com/java/ref_string_endswith.asp) | Checks whether a string ends with the specified character  (s) | boolean |
| [equals()](https://www.w3schools.com/java/ref_string_equals.asp) | Compares two strings. Returns true if the strings are equal,  and false if not | boolean |
|  |  |  |
| [equalsIgnoreCase()](https://www.w3schools.com/java/ref_string_equalsignorecase.asp) | Compares two strings, ignoring case considerations | boolean |

12)How can a java string can be converted into an byte array?

Ans:

13)What is inheritance in java?

Ans: Inheritance in Java is **a concept that acquires the properties from one class to other classes**; for example, the relationship between father and son. Inheritance in Java is a process of acquiring all the behaviours of a parent object.

14)What is super class and subclass in java?

Ans: Definitions: **A class that is derived from another class is called a subclass** (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class).

15)Which class in java is super class of every other class ?

Ans: **Object class** is the root or superclass of the class hierarchy, which is present in java. lang package. All predefined classes and user-defined classes are the subclasses from Object class.

16)Types of inheritance in java?

* Ans: Single-level inheritance.
* Multi-level Inheritance.
* Hierarchical Inheritance.
* Multiple Inheritance.
* Hybrid Inheritance.

17)Can a class extend more than one class?

Ans:No a class cannot extend more than one class because multiple inheritance is not used in java in case of class.If we try to use then ambiguity problem will be created.

18)Can we extend final class?

Ans: The main purpose of using a class being declared as final is to prevent the class from being subclassed. If a class is marked as final then no class can inherit any feature from the final class. **You cannot extend a final class**.

19) Why multiple inheritance is not supported in java?

Ans: Java doesn't support multiple inheritances in classes because **it can lead to diamond problem** and rather than providing some complex way to solve it, there are better ways through which we can achieve the same result as multiple inheritances.

20)What is an interface in java?

Ans: 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.

21)Can we define an interface with a static modifier?

Ans: Similar to Default Method in Interface, **the static method in an interface can be defined in the interface**, but cannot be overridden in Implementation Classes. To use a static method, Interface name should be instantiated with it, as it is a part of the Interface only.

22)Can an interface extend another interface in java?

Ans: **An interface can extend other interfaces**, just as a class subclass or extend another class. However, whereas a class can extend only one other class, an interface can extend any number of interfaces.

23)Why an interface can extend more than one interface but a class cannot extend more than one class?

Ans: The interface is a pure abstraction model and does not have inheritance hierarchy like classes that's why interface be able to extend more than one interface but a class can't extend more than one class because **Java doesn't allow multiple inheritance, so the class is restricted to extend only one class**.

24)what is difference between abstract class and interface in java?

Ans:

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8,  it can have **default and static methods** also. |
|  |  |
| 2) Abstract class **doesn't support multiple inheritance**. | Interface **supports multiple inheritance**. |
|  |  |
| 3) Abstract class **can have final, non-final, static and non-static variables**. | Interface has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of**  **abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword  "implements". |
| 8) A Java **abstract class** can have class members like private, protected, etc. | Members of a Java interface are public by default. |
| 9)**Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Drawable{ void draw(); } |

25)Define package in java ?

Ans: A **java package** is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

26) Which package is always imported by default?

Ans: **lang Package**. Java compiler imports java. lang package internally by default. It provides the fundamental classes that are necessary to design a basic Java program.

27)What are the benefits of using multithreading?

Ans: **Benefits of Multithreading\***

* Improved throughput. ...
* Simultaneous and fully symmetric use of multiple processors for computation and I/O.
* Superior application responsiveness. ...
* Improved server responsiveness. ...
* Minimized system resource usage. ...
* Program structure simplification. ...
* Better communication.

28)What is thread in java?

Ans: A thread is **a thread of execution in a program**. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently. Every thread has a priority. Threads with higher priority are executed in preference to threads with lower priority.

29)What are the ways of implementing the thread in java?

Ans: There are two ways to create a thread: **By extending Thread class**. **By implementing Runnable interface**.

30)What are the wait() and sleep() methods in java?

Ans:

|  |  |  |
| --- | --- | --- |
| 1. | The Wait() method is related to the Object class. | The Sleep () method is related to the Thread class. |
| 2. |  | The Sleep () method does not release the lock on  the object during Synchronization. |
| 3. | It is not a static method. | It is a static method. |
| 4. | At the time of the Synchronization, the Wait() method releases obj. | At the time of the Synchronization, the Sleep() method  doesn't release the obj, i.e., lock. |
| 5. | We can call the Wait () method only from the Synchronized context. | We can call the Sleep () method from outside the  Synchronized context. |
| 6. | The Sleep() method has two overloaded methods, which are as follows:   * sleep(long milliseconds, int nanoseconds) * sleep(long milliseconds) | The Sleep() method has three overloaded methods,  which are as follows:   * Wait() * wait(long timeout, int nanoseconds) * wait(long timeout) |
| 7. | The constructor of the Wait() method is defined in the following way: public final void Wait(long timeout) | The constructor of the Sleep () method  In the following way: public static void Sleep (long Millis) throws  InterruptedExecption |

31)What is the difference bet notify and notifyAll() methods?

Ans:

| **Sr. No.** | **Key** | **notify** | **notifyAll** |
| --- | --- | --- | --- |
| 1 | Notification | In case of multiThreading notify() method sends the notification to only one thread among the multiple waiting threads which are waiting for lock. | While notifyAll() methods in the same context sends the notification to all waiting threads instead of single one thread. |
| 2 | Thread identification | As in case of notify the notification is sent to single thread among the multiple waiting threads so it is sure that which of those waiting thread is going to receive the lock. | On other hand notifyAll sends notification  to all waiting threads hence it is not clear which of the thread is going to receive the lock. |
| 3 | Risk factor | In case of notify() method the risk of thread missing is high as notification is sent only single thread and if it misses that than no other thread would get notification and hence the lock. | While in case of notifyAll as notification is  to all the waiting threads and hence if any thread misses the notification, there are other threads to do the job. Hence risk is less. |
| 4 | Performance | Memory and CPU drain is less as compare to notifyAll as notification is sent to single one thread so performance is better as compare to notifyAll. | On other hand as the cost of no notification is dropped and notification is sent to all waiting threads the memory and CPU  drain is more as compare to notify and  hence performance of notifyAll is lesser. |
| 5 | Interchangeable | In case of the notify() method as only single one thread is in picture hence no concept of thread Interchangeable is possible. | While we should go for notifyAll() if all  your waiting threads are interchangeable (the order they wake up doesn’t   matter). |

32)What wait(),notify() and notifyAll() methods are present in object class?

Ans: If wait() and notify() were on the Thread instead then each thread would have to know the status of every other thread and there is no way to know thread1 that thread2 was waiting for any resource to access. Hence, notify, wait, notifyAll methods are defined in object class in Java.

33)What is the start and run() method of thread class?

Ans: **start method of thread class is implemented as when it is called a new Thread is created and code inside run() method is executed in that new Thread**. While if run method is executed directly than no new Thread is created and code inside run() will execute on current Thread and no multi-threading will take place.

34)What is the purpose of the join method?

Ans: Join method in Java **allows one thread to wait until another thread completes its execution**. In simpler words, it means it waits for the other thread to die.

35)Explain life cycle of thread?

36)How do thread communicate with each other?

Ans: **Inter-thread communication i**nvolves the communication of threads with each other. The three methods that are used to implement inter-thread communication in Java

## wait()

This method causes the current thread to**release the lock**. This is done until a specific amount of time has passed or another thread calls the **notify()** or **notifyAll()**method for this object.

## notify()

This method **wakes a single thread** out of multiple threads on the current object’s monitor. The choice of thread is arbitrary.

## notifyAll()

This method **wakes up all the threads** that are on the current object’s monitor.

37)What is synchronization process? why use it?

Ans: Synchronization in java is **the capability to control the access of multiple threads to any shared resource**. In the Multithreading concept, multiple threads try to access the shared resources at a time to produce inconsistent results. The synchronization is necessary for reliable communication between threads.

38)What is synchronization method and synchronization block? Which one should be preferred?

Ans: **synchronized block has better performance as only the critical section is locked but synchronized method has poor performance than block**. synchronized block provide granular control over lock but synchronized method lock either on current object represented by this or class level lock.

39)What is mean by exception handling?

Ans: 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**.

40)what is the difference between exception and error?

Ans:

| **Sr. No.** | **Key** | **Error** | **Exception** |
| --- | --- | --- | --- |
| 1 | Type | Classified as an unchecked type | Classified as checked and unchecked |
| 2 | Package | It belongs to java.lang.error | It belongs to java.lang.Exception |
| 3 | Recoverable/ Irrecoverable | It is irrecoverable | It is recoverable |
| 4 |  | It can't be occur at compile time | It can occur at run time compile  time both |
| 5 | Example | OutOfMemoryError ,IOError | NullPointerException , SqlException |

41)Name the different types of exception in java?

### Ans: **Checked Exception**

**Checked** exceptions are called **compile-time** exceptions because these exceptions are checked at compile-time by the compiler. The compiler ensures whether the programmer handles the exception or not. The programmer should have to handle the exception; otherwise, the system has shown a compilation error.

### **Unchecked Exceptions**

The **unchecked** exceptions are just opposite to the **checked** exceptions. The compiler will not check these exceptions at compile time. In simple words, if a program throws an unchecked exception, and even if we didn't handle or declare it, the program would not give a compilation error. Usually, it occurs when the user provides bad data during the interaction with the program.

42)Can we just use try block instead finally and catch block?

43)Difference between checked and unchecked exception?

Ans:

|  |  |
| --- | --- |
| Checked exceptions occur at compile time. | Unchecked exceptions occur at runtime. |
| The compiler checks a checked exception. | The compiler does not check these types of exceptions. |
| These types of exceptions can be handled at the time of compilation. | These types of exceptions cannot be a catch or handle at  the time of compilation, because they get generated by the  mistakes in the program. |
| They are the sub-class of the exception class. | They are runtime exceptions and hence are not a part of the  Exception class. |
| Here, the JVM needs the exception to catch and handle. | Here, the JVM does not require the exception to catch and  handle. |
| Examples of Checked exceptions:   * File Not Found Exception * No Such Field Exception * Interrupted Exception * No Such Method Exception * Class Not Found Exception | Examples of Unchecked Exceptions:   * No Such Element Exception * Undeclared Throwable Exception * Empty Stack Exception * Arithmetic Exception * Null Pointer Exception * Array Index Out of Bounds Exception * Security Exception |

44)What is mean by throws keyword?

Ans: The throws keyword in Java is **used to declare exceptions that can occur during the execution of a program**. For any method that can throw exceptions, it is mandatory to use the throws keyword to list the exceptions that can be thrown.