1 Explain instanceof operator.

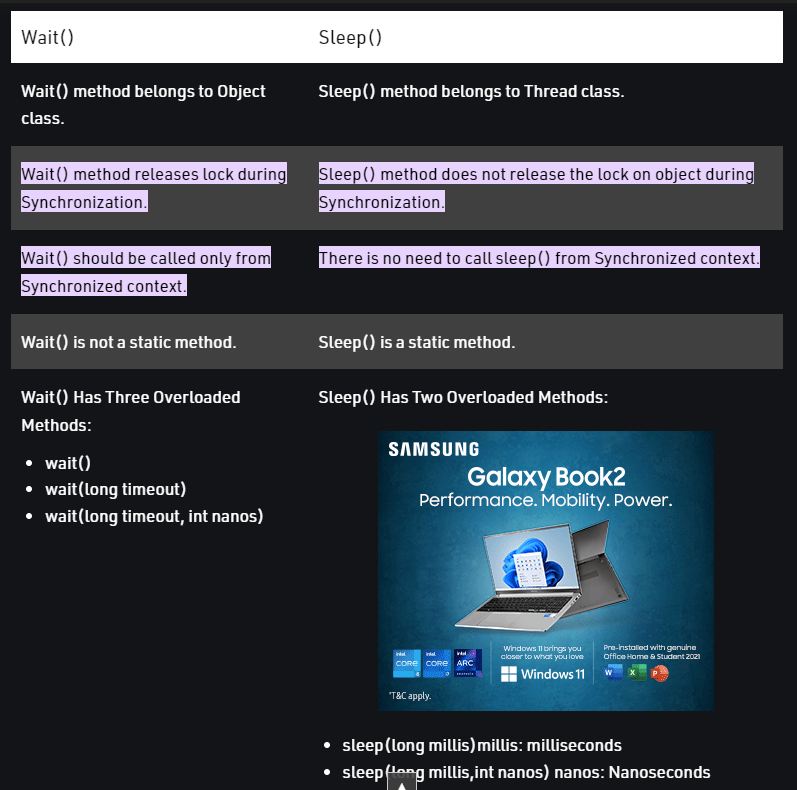
A. is used to check whether an object is an instance of a particular class or not. Its syntax is. objectName instanceOf className; Here, if objectName is an instance of className , the operator returns true . Otherwise, it returns false .

2 There is no destructor in Java. Justify.

ANS. It is a special method that automatically gets called when an object is no longer used. When an object completes its life-cycle the garbage collector deletes that object and deallocates or releases the memory occupied by the object.

Java has its own garbage collection implementation so it does not require any destructor like C++ . This makes Java developer lazy in implementing memory management.

3 Compare methods wait and sleep.



4 Explain method join with the help of an example.

1. **join():**It will put the current thread on wait until the thread on which it is called is dead. If thread is interrupted then it will throw InterruptedException.  
   **Syntax:**
2. public final void join()
3. **join(long millis)**:It will put the current thread on wait until the thread on which it is called is dead or wait for specified time (milliseconds).  
   **Syntax:**
4. public final synchronized void join(long millis)
5. **join(long millis, int nanos):**It will put the current thread on wait until the thread on which it is called is dead or wait for specified time (milliseconds + nanos).  
   **Syntax:**
6. public final synchronized void join(long millis, int nanos)

5 Explain two usage of keyword synchronized.

Synchronized method is used **to lock an object for any shared resource**. When a thread invokes a synchronized method, it automatically acquires the lock for that object and releases it when the thread completes its task.

6 JVM is platform dependent. Justify.

JVM is platform dependent **because it takes java byte code and generates byte code for the current operating system**. So Java software is platform dependent but Java language is platform independent because different operating system have different JVMs.

7 Can array be initialized at the time they are defined? If yes, explain its syntax with

suitable example.

int[] arr={ 1,2,3,4};

8 What are multidimensional arrays? Explain their syntax and mechanism for accessing

their elements.

**Multidimensional Arrays** can be defined in simple words as array of arrays. Data in multidimensional arrays are stored in tabular form (in row major order).

int[][] arr={{1,2},{1,4}};

**Accessing Elements of Two-Dimensional Arrays**

Elements in two-dimensional arrays are commonly referred by **x[i][j]** where ‘i’ is the row number and ‘j’ is the column number.

int[][] arr = new int[10][20];

arr[0][0] = 1;

9 What is class? Describe the syntax for declaring a class with an example.

a **class** is a basic building block. Java provides a reserved keyword **class** to define a class. The keyword must be followed by the class name. Inside the class, we declare methods and variables

10 What are objects? Describe the syntax for declaring a objects with an example.

It is a basic unit of Object-Oriented Programming and represents real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of :

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

11 Explain how java supports encapsulation and data abstraction.

Abstraction is a feature of OOPs that hides the **unnecessary** detail but shows the essential information.

Encapsulation is also a feature of OOPs. It hides the code and data into a **single** entity or unit so that the data can be protected from the outside world.

12 What are access modifier? Illustrate with an example.

In Java, access modifiers are **used to set the accessibility (visibility) of classes, interfaces, variables, methods, constructors, data members, and the setter methods**.

For example, class Animal { public void method1() {...} private void method2() {...} }



13 Discuss parameter passing by value and by reference in Java with suitable example.

14 What role do interface play in java?

### Why do we use an Interface?

* It is used to achieve total abstraction.
* Since java does not support multiple inheritances in the case of class, by using an interface it can achieve multiple inheritances.
* It is also used to achieve loose coupling.
* Interfaces are used to implement abstraction. So the question arises why use interfaces when we have abstract classes?

15 How can interface be used to support multiple inheritance ?(Give an example)

16 What are packages in Java?

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.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

17 How do we design, create, and access a package? Discuss with an example

1. import package.\*;
2. import package.classname;
3. fully qualified name.

1 Explain characteristics of OOP ?

## **What are the Objects in OOPs ( Object-oriented programming system )?**

* Objects are basic building blocks for designing programs.
* An object is a collection of data members and associated member functions.
* An object may represent a person, place or a table of data.
* Each object is identified by a unique name. Each object must be a member of a particular class.
* Example: chair, table, whiteboard are the objects of the class (class).

## **What are the Classes in OOPs (Object-oriented programming system)?**

* The objects can be made user-defined data types with the help of a class.
* A class is a collection of objects that have identical properties, common behaviour and shared relationship.
* Once the class is defined any number of objects of that class is created.
* Classes are user-defined data types A class can hold both data and functions.
* For example planets, sun and moon are the members of the solar system class.

## **What is Data abstraction in OOPs (Object-oriented programming system)?**

Data abstraction refers to the process of representing essential features without including background details or explanations.

## **What is Data encapsulation in OOPs (Object-oriented programming system)?**

* The wrapping of data and function into a single unit is called data encapsulation.
* Data encapsulation enables data hiding and information hiding.

## **What is Inheritance in OOPs (Object-oriented programming system)?**

* Inheritance is the process by which one object can acquire and the use of properties of another object.
* The existing class is known as a base class or superclass.
* The new class is known as a derived class or subclass.
* The derived class shares some of the properties of the base class. Therefore a code from a base class can be reused by a derived class.

## **What is Polymorphism in OOPs (Object-oriented programming system)?**

* The ability of an operator and function to take.
* Multiple forms are known as polymorphism.
* The different types of polymorphism are operator
* overloading and function overloading.

2 List various features of Java? Also explain any two feature with example.

### Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun Microsystem, Java language is a simple programming language because:

* Java syntax is based on C++ (so easier for programmers to learn it after C++).
* Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
* There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

### Platform Independent



Java is platform independent because it is different from other languages like [C](https://www.javatpoint.com/c-programming-language-tutorial), [C++](https://www.javatpoint.com/cpp-tutorial), etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on top of other hardware-based platforms. It has two components:

1. Runtime Environment
2. API(Application Programming Interface)

Java code can be executed on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere (WORA).

3 Explain Dynamic Method Dispatch with example

**Dynamic method dispatch** is the mechanism in which a call to an overridden method is resolved at run time instead of compile time. This is an important concept because of how Java implements run-time polymorphism.

// Implementing Dynamic Method Dispatch

class Apple

{

void display()

{

System.out.println("Inside Apple's display method");

}

}

class Banana extends Apple

{

void display() // overriding display()

{

System.out.println("Inside Banana's display method");

}

}

class Cherry extends Apple

{

void display() // overriding display()

{

System.out.println("Inside Cherry's display method");

}

}

class Fruits\_Dispatch

{

public static void main(String args[])

{

Apple a = new Apple(); // object of Apple

Banana b = new Banana(); // object of Banana

Cherry c = new Cherry(); // object of Cherry

Apple ref; // taking a reference of Apple

ref = a; // r refers to a object in Apple

ref.display(); // calling Apple's version of display()

ref = b; // r refers to a object in Banana

ref.display(); // calling Banana's version of display()

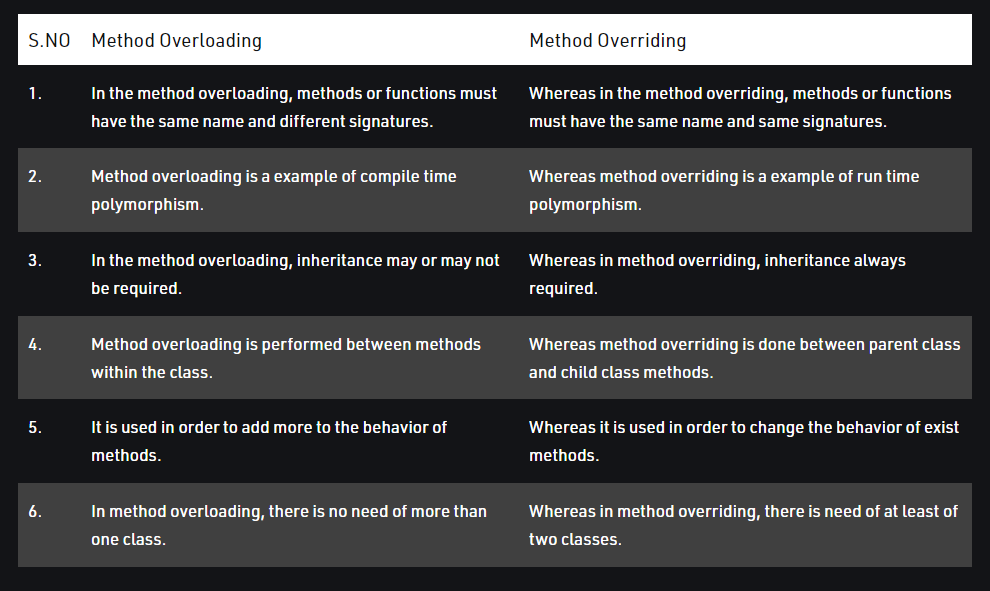
ref = c; // r refers to a object in Cherry

ref.display(); // calling Cherry's version of display()

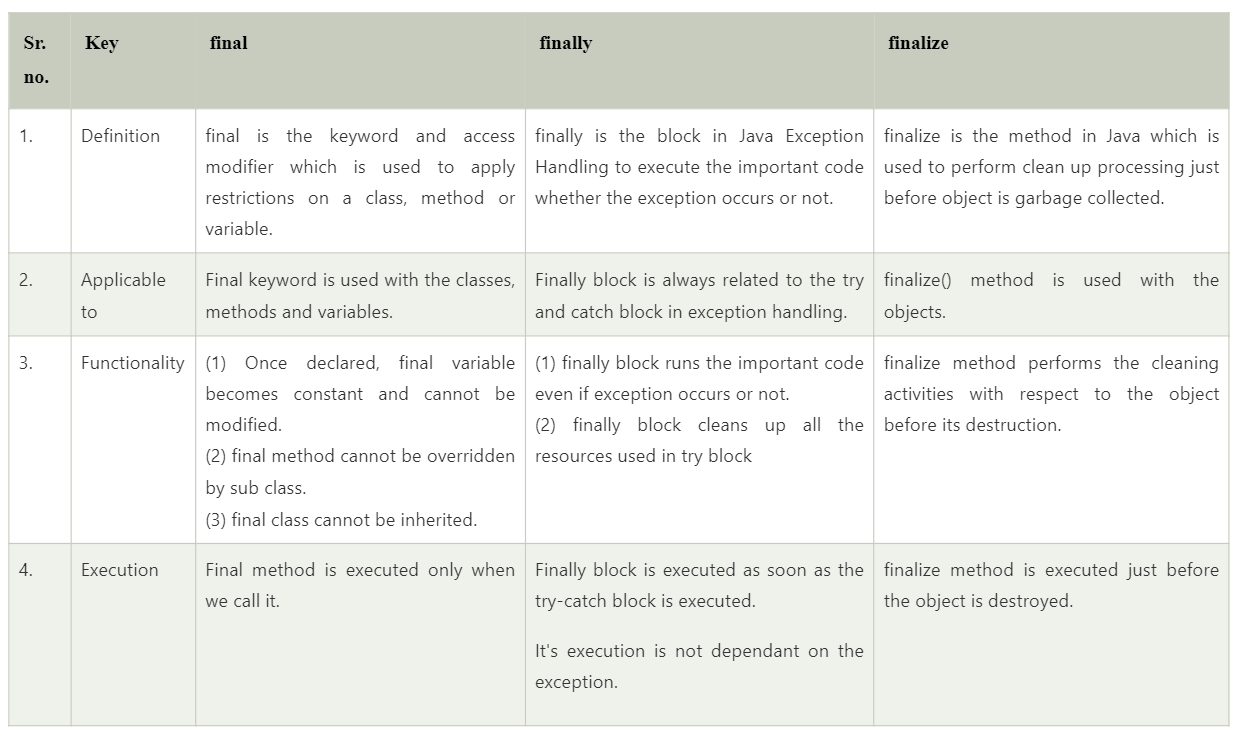
}

}

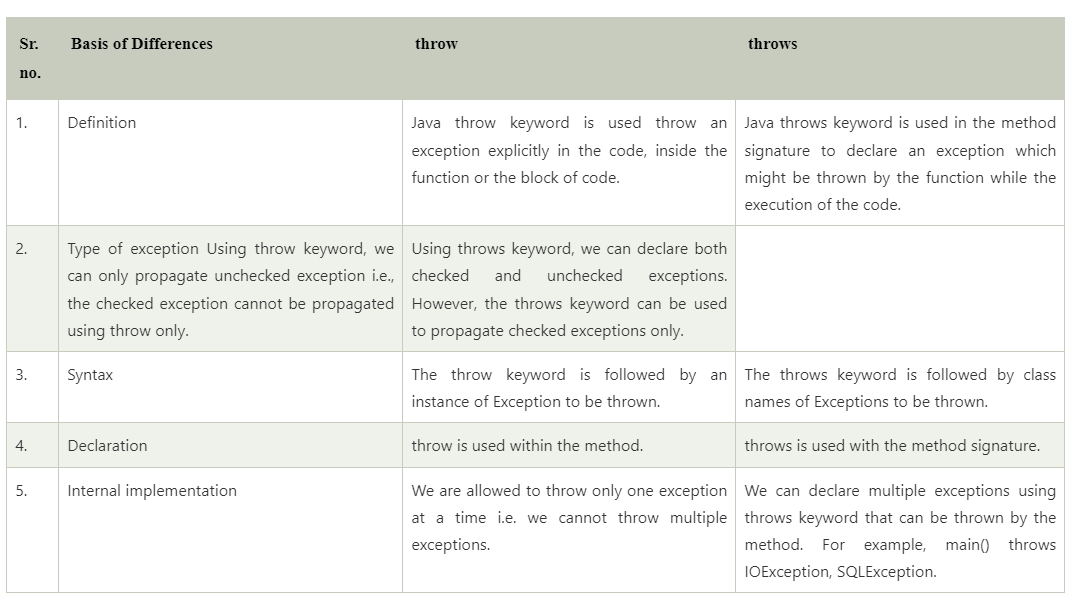
4 Difference between method overloading and method overriding with suitable examples.

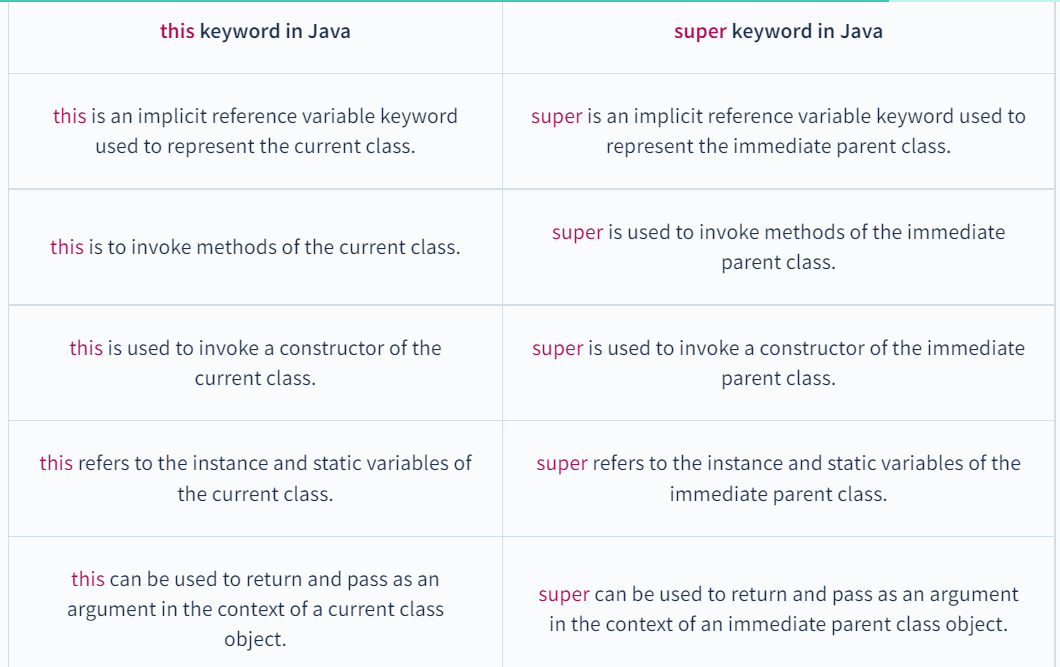


5 Explain : (1) Final (2) Finally (3) Finalize



6 Explain (1) Static (2) Super (3) throw (4) throws (5)this



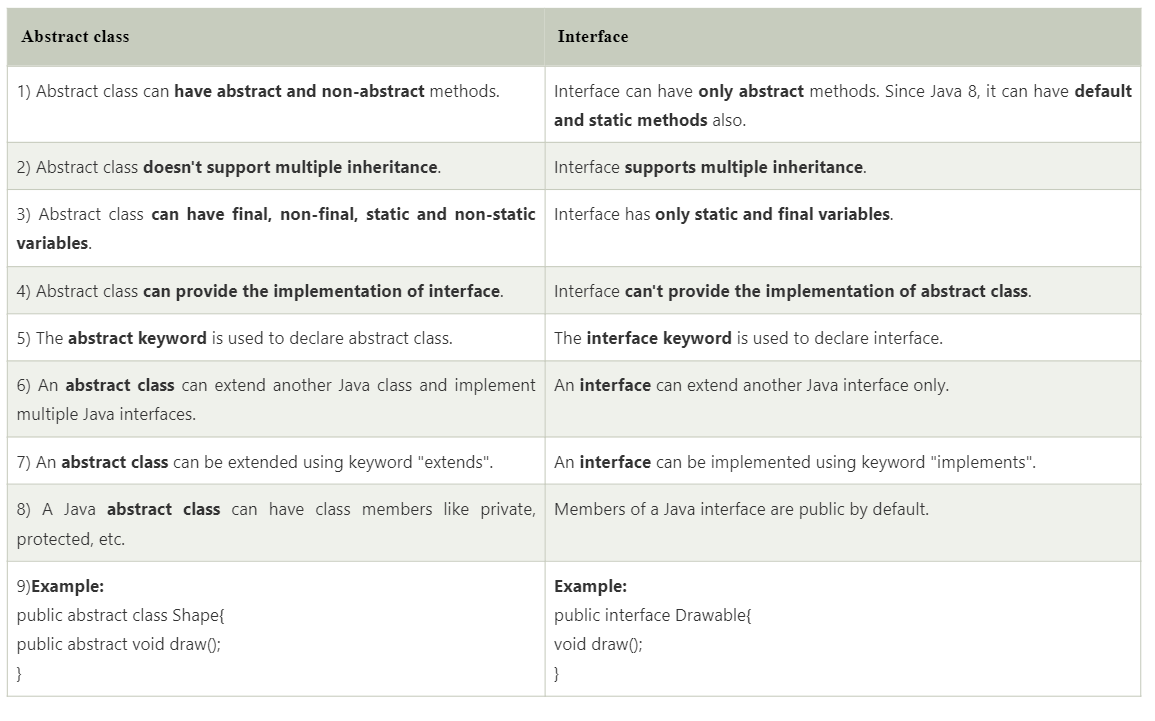


The **static keyword** in [Java](https://www.javatpoint.com/java-tutorial) is used for memory management mainly. We can apply static keyword with [variables](https://www.javatpoint.com/java-variables), methods, blocks and [nested classes](https://www.javatpoint.com/java-inner-class). The static keyword belongs to the class than an instance of the class.

The static can be:

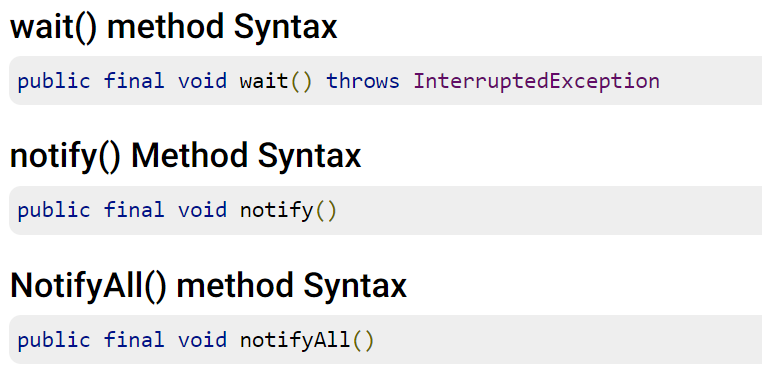
1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block
4. Nested class

7 Differentiate Abstract class and interface with suitable example.

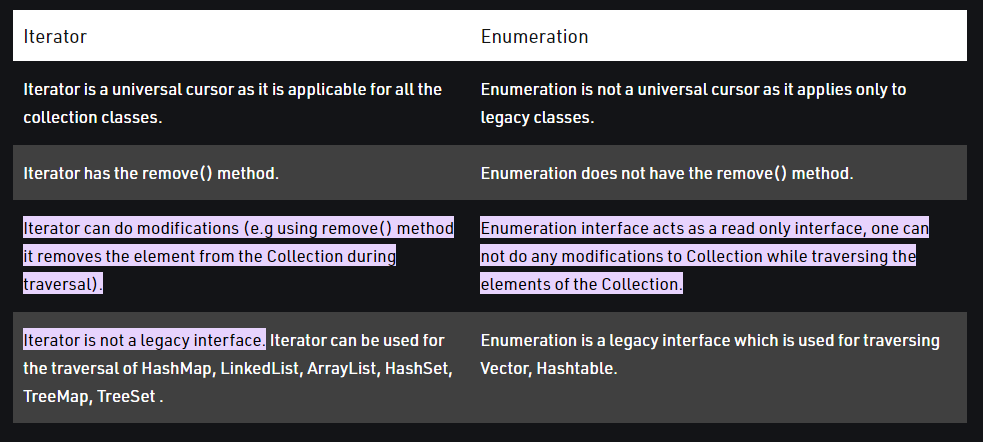


8 Explain Life cycle of Thread. Describe wait(), notify() and notifyall() and synchronized methods in thread.

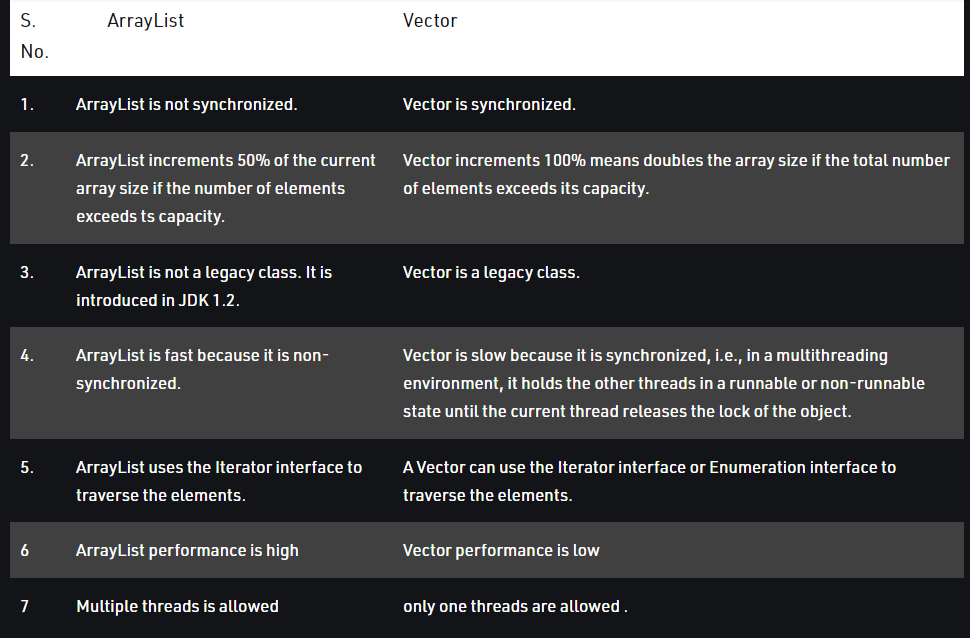
These are **final**methods defined in the **Object**class and can be called only from within a **synchronized**context. The **wait()** method causes the current thread to wait until another thread invokes the **notify()**or **notifyAll()**methods for that object. The **notify()** method **wakes up a single thread** that is waiting on that object’s monitor. The **notifyAll()** method **wakes up all threads** that are waiting on that object’s monitor.



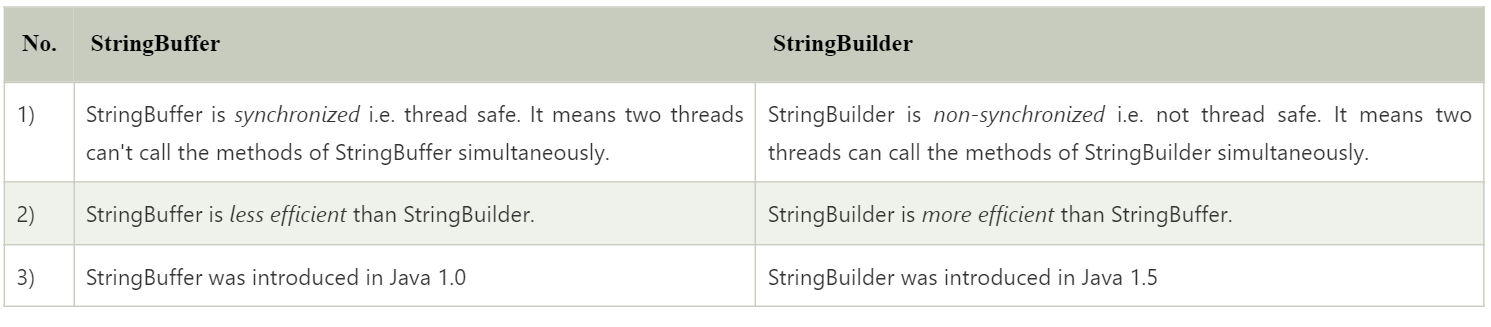
9 Differentiate Enumeration and Iterator.



Vector and Array



String class and StringBuffer class, StringBuffer and String Builder class



10 What is inheritance? Explain the need of inheritance with suitable example.

11 Discuss various forms of inheritance with example.

12 Define the following OO related terms: a. Dynamic Binding b. Polymorphism

13 What are exception and how are they handled in Java? Give an example.

14 What is Exception? Explain various Built-in exceptions in java. Also give difference between throw and throws keywords.

15 Explain package in java. List out steps to create user defined package with one example.

16 Define generics in java. Write a program to demonstrate generic interface and generic method. 17 What is collection in Java? Differentiate between Vector and ArrayList.

18 What is Wrapper class in Java? Explain with examples.

19 What are the benefits of using generic types?