**Java basics**

* **Which language are both interpreted and compiled?**

Java can be considered both a compiled and an interpreted language because its source code is first compiled into a binary byte-code. This byte-code runs on the Java Virtual Machine (JVM), which is usually a software-based interpreter.

* **Name the primitive types in Java.**

Primitive data types - includes **byte , short , int , long , float , double , boolean** and **char**.

* **What is the difference between primitive and non-primitive data types?**

Non-primitive data types are called **reference types** because they refer to objects.

The main difference between **primitive** and **non-primitive** data types are:

* Primitive types are predefined (already defined) in Java. Non-primitive types are created by the programmer and is not defined by Java (except for String).
* Non-primitive types can be used to call methods to perform certain operations, while primitive types cannot.
* A primitive type has always a value, while non-primitive types can be null.
* A primitive type starts with a lowercase letter, while non-primitive types starts with an uppercase letter.
* The size of a primitive type depends on the data type, while non-primitive types have all the same size.

Examples of non-primitive types are [Strings](https://www.w3schools.com/java/java_strings.asp), [Arrays](https://www.w3schools.com/java/java_arrays.asp), [Classes,](https://www.w3schools.com/java/java_classes.asp)[Interface](https://www.w3schools.com/java/java_interface.asp), etc.

* **Does Java make distinction between value type and reference type?**

*In Java, all objects and enums are reference types, and all primitives are value types.* The distinction between the two is the same as in C# with respect to copy semantics, but you cannot define a new value type in Java.

* **If we don't override hashCode and override only equals then what happen? -** <https://www.geeksforgeeks.org/equals-hashcode-methods-java/>
* **What is the difference between reference types and value types?**
* **What are the uses of final keyword in Java?**

Class: used to stop inheriting a class from another method

Method: used to stop overriding a method

Field: used to declare constant (value can’t be changed once set)

* **What is the difference between jar and war files?**

https://stackoverflow.com/questions/5871053/difference-between-jar-and-war-in-java

* **What are annotations? What are some of the annotations in java.lang package?**

<https://www.geeksforgeeks.org/annotations-in-java/>

* **Can we override/overload static method?**

<https://www.geeksforgeeks.org/can-we-overload-or-override-static-methods-in-java/>

* **Can we override private method?**

<https://www.geeksforgeeks.org/can-override-private-methods-java/#:~:text=2)%20In%20Java%2C%20methods%20declared,fact%20bounded%20during%20compile%20time>.

* **What is the difference between == and equals() in Java?**

<https://stackoverflow.com/questions/7520432/what-is-the-difference-between-and-equals-in-java>

<https://www.geeksforgeeks.org/difference-between-and-equals-method-in-java/>

== is a reference comparison, i.e. both objects point to the same memory location

.equals() evaluates to the comparison of values in the objects

* **Can we reference instance variable from static method? if not why?** <https://www.geeksforgeeks.org/java-program-to-check-the-accessibility-of-an-instance-variable-by-a-static-method/>
* **What are the different ways of reusing code in JAVA?**

Inheritance, Aspect Oriented Programming

* **Will a finally block execute after a return statement in a method in Java?** https://www.tutorialspoint.com/will-a-finally-block-execute-after-a-return-statement-in-a-method-in-java#:~:text=Yes%2C%20the%20finally%20block%20will,it%20will%20not%20be%20executed.
* **How to get an enum value from a string value in Java?**

<https://stackoverflow.com/questions/604424/how-to-get-an-enum-value-from-a-string-value-in-java?rq=1>

* **What is the meaning of public, static, void?**

<https://www.geeksforgeeks.org/replacing-public-with-private-in-main-in-java/?ref=lbp>

* **What is the difference between public, protected, package-private and private in Java?**

[https://stackoverflow.com/questions/215497/what-is-the-difference-between-public- protected-package-private-and-private-in?rq=1](https://stackoverflow.com/questions/215497/what-is-the-difference-between-public-%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20protected-package-private-and-private-in?rq=1)

**What will be the output of below code?**

a)

public class Test { public static void main(String[] args) {

System.out.println(Math.min(Double.MIN\_VALUE, 0.0d)); } }

b)

public class Test{

public static void main(String[] arr){

System.out.println(0.1\*3 == 0.3);

System.out.println(0.1\*2 == 0.2);

}

}

c)

public class Test

{

public static void main(String[] args)

{

System.out.println(1.0 / 0.0);

System.out.println(1 / 0);

}

}

**Arrays**

Variable Args in Java

Variable Array In java

**What is the difference between Array and ArrayList?**

**Strings**

* **Why is string immutable?**
* **Why is string a reference type?**
* **What is the minimum iteration needed to reverse the String - String Swap.**
* **Write a program to split a string and find out word count.**
* **String permutations and combinations example "abcde" create possible P & C**
* **How many objects will be created?**

public class Test{

public static void main(String[] args){

String s = new String("Hello World"); //

}

}

**OOPS concepts**

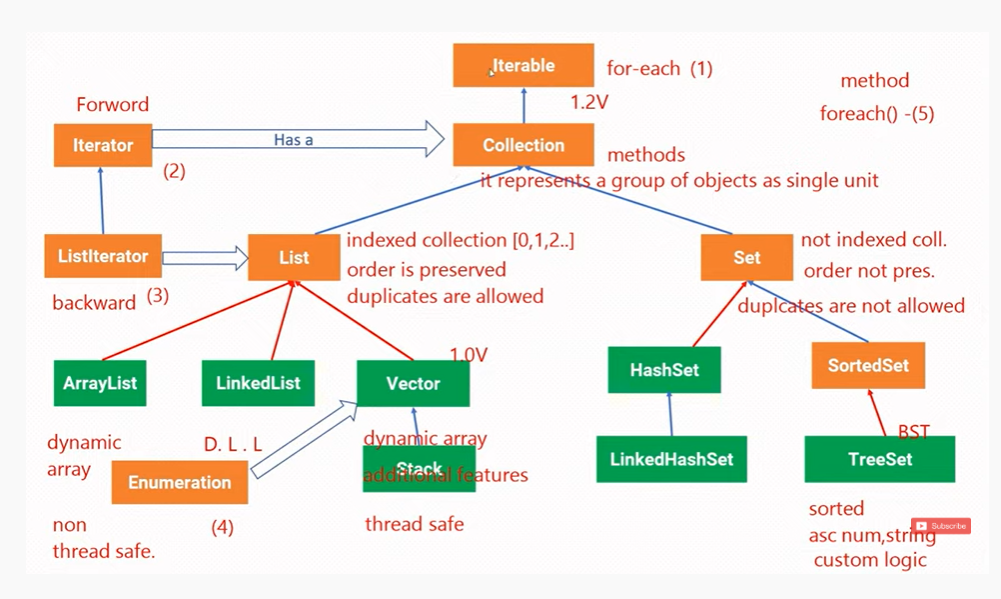
* **Can we have private constructors? What are the uses of private constructors?**
* **Like any method we can provide access specifier to the constructor. If it’s made private, then it can only be accessed inside the class. There are**
* **Various scenarios where we can use private constructors. The major ones are**
* **Internal Constructor chaining**
* **Singleton class design pattern.**
* **Implement singleton in a thread safe way.**
* **Why Java does not support multiple inheritance? What is the diamond ring problem?**
* **What is eager and lazy loading? What is dynamic method dispatch?**

**Generics**

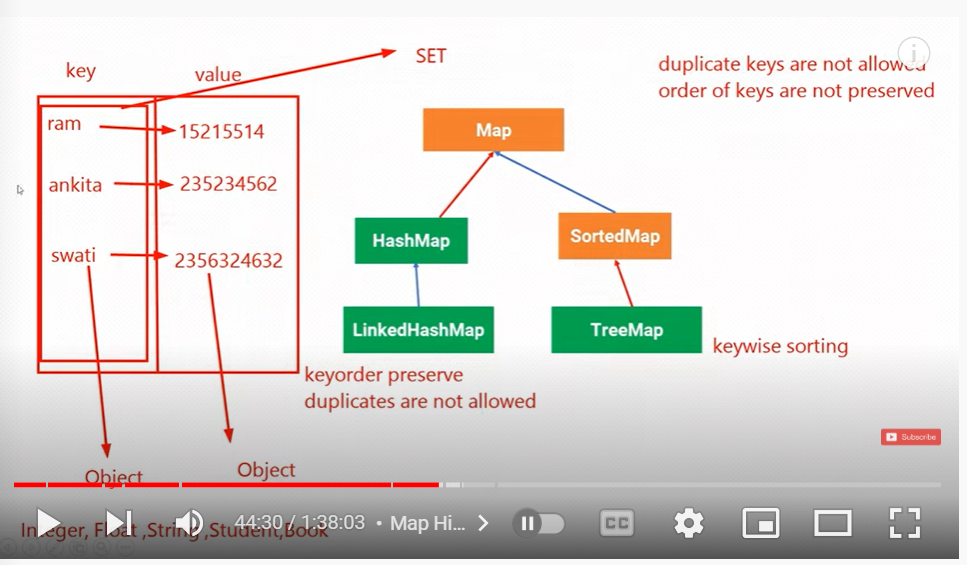
* generic vs Inheritance
* Why Generics

**Collections**

* **Draw the diagram of collections hierarchy.**

****

* **Draw class hierarchy of Map.**

****

* **Number addition and subtraction in parallel streams**
* **Methods in streams**
* **2 level sorting in java**
* **How to create a Ordered HashSet - LinkedHashSet**
* **Linked List vs Array**
* **How to create your own concurrentHashMap**
* **Implement hashtable/HashMap in java**
* **bracket Problem in java - Stack**
* **What will happen if we add another element with same key in hashmap?**
* <https://stackoverflow.com/questions/1669885/what-happens-when-a-duplicate-key-is-put-into-a-hashmap>
* **What is hashing? How do we override hashcode? What are the things to be taken care while implementing hashcode method?**
* **Which linked list is used for HashMap implementation in Java? Is it singly or doubly linked list?** <https://www.geeksforgeeks.org/hashtables-chaining-with-doubly-linked-lists/>
* **If we have 2 employee objects with same id, name and DOB, how will you store in HashSet?** <https://www.codexpedia.com/java/java-set-and-hashset-with-custom-class/>
* **Will it store both or 1 object in HashSet? Do we have to do anything in Employee object for this?**
* **Given an array, write code to return new array with numbers less than k?**

Use streams api – filter

* **How HashMap works internally? If you've any custom object like Employee and you want to use that as key in hashmap, which two methods you will override?**
* **What do you need to do to use a custom object as key in Collection classes like Map or Set?**If you are using any custom object in Map as key, you need to override equals() and hashCode() method, and make sure they follow there contract. On the other hand if you are storing a custom object in Sorted Collection e.g. SortedSet or SortedMap, you also need to make sure that your equals() method is consistent to compareTo() method, otherwise those collection will not follow there contacts e.g. Set may allow duplicates.
* **How do you order data stored in hashmap by key and by value?**
* **How do you copy unique values from hashmap into arraylist?**
* **How do you design custom collection and it should be iterable? Iterable vs Interator.**
* **What is the difference between HashMap and TreeMap?**
* **What are hash collisions? How do you avoid hash collisions?**
* **How do you remove an entry from a Collection? and subsequently what is the difference between the remove() method of Collection and remove() method of Iterator, which one you will use while removing elements during iteration?**  
  <https://javarevisited.blogspot.com/2011/11/collection-interview-questions-answers.html#axzz7jmzCQQTz>

Read more: <https://javarevisited.blogspot.com/2011/11/collection-interview-questions-answers.html#ixzz7jslNDeol>

* **When to use LinkedList over ArrayList in Java?**

<https://stackoverflow.com/questions/322715/when-to-use-linkedlist-over-arraylist-in-java?rq=1>

* **What is the difference between HashTable abd HashMap?**

**Concurrency**

* **What is difference between synchronized and re-entrant lock?**

<https://stackoverflow.com/questions/9072422/difference-between-synchronized-and-re-entrant-lock>

<https://www.geeksforgeeks.org/reentrant-lock-java/>

* **BlockingQueue and Semaphores**
* **2 thread of same object accessing same non static method (synchronized).**
* **What is the difference between ConcurrentHashMap and HashMap?**

<https://stackoverflow.com/questions/510632/whats-the-difference-between-concurrenthashmap-and-collections-synchronizedmap>

* **Why it is preferred to implement Runnable instead of extending Thread class?**

<https://stackoverflow.com/questions/541487/implements-runnable-vs-extends-thread-in-java>

**Lambda and functional & reactive programming**

* **Reactive programming**
* **What is functional interface? What is the difference between normal interface and functional interface?**
* **Argument of Predicate**
* **Name some of the functional interface**
* **Predicate functional interface method and argument**
* **What do you mean by captured variable?**

<https://www.geeksforgeeks.org/java-lambda-expression-variable-capturing-with-examples/>

* **What are method references in Java?**

https://www.geeksforgeeks.org/method-references-in-java-with-examples/ <https://www.javatpoint.com/java-8-method-reference>

* **What are Intermediate operations (methods) in Java?**
* **What is the difference between intermediate and terminal operations?**

<https://stackoverflow.com/questions/47688418/what-is-the-difference-between-intermediate-and-terminal-operations>

* **What is the difference between map & flatmap?**
* **What is the use of @FunctionalInterface, custom functional interface?**
* **Name some of the built in functional interface.**

**Streams**

https://www.geeksforgeeks.org/foreach-loop-vs-stream-foreach-vs-parallel-stream-foreach/?ref=rp

**Garbage collections**

* **Garbage collector type and details**
* **How Mark and Sweep works**
* **Parts of Heap Memory (JVM Architecture)**
* **What are memory leaks and garbage collection?**