Java general questions:

1. What is marker interface? Can we create user defined marker interface?

https://www.geeksforgeeks.org/marker-interface-java/

1. Serialization and deserialization in java

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

1. How many ways we can break singleton and what is the way to prevent them.

Reflection, Serialization and Cloning can break the Singleton pattern.

<https://www.geeksforgeeks.org/prevent-singleton-pattern-reflection-serialization-cloning/>

1. When class load exception will occur in java?

If the class which we want to use not present in the classpath or all dependent class is not present in the containing jar and same class we are using in our programme.

1. Can abstract class have public constructor?

**Yes**! **Abstract classes can have constructors**!

Yes, when we define a class to be an Abstract Class it cannot be instantiated but that does not mean an Abstract class cannot have a constructor. Each abstract class must have a concrete subclass which will implement the abstract methods of that abstract class.

When we create an object of any subclass all the constructors in the corresponding inheritance tree are invoked in the top to bottom approach (also known as constructor chaining). The same case applies to abstract classes. Though we cannot create an object of an abstract class, when we create an object of a class which is concrete and subclass of the abstract class, the constructor of the abstract class is automatically invoked. Hence we can have a constructor in abstract classes.

Note: A non-abstract class cannot have abstract methods but an abstract class can have a non-abstract method. Reason is similar to that of constructors, difference being instead of getting invoked automatically we can call super(). Also, there is nothing like an abstract constructor as it makes no sense at all.

Public constructor in the abstract class make no sense as we need to call these constructor always from the derived class or concrete class, so better we can use protected access modifier.

**Java 8 feature programming example link**

Static and default method have been introduced in the Java 8.

//trying to override Object method gives compile time error as

//"A default method cannot override a method from java.lang.Object"

// default String toString(){

// return "i1";

// }

<https://www.journaldev.com/2752/java-8-interface-changes-static-method-default-method>.

We can instantiate an interface with the anonymous class

Runnable r = new Runnable(){

@Override

public void run() {

System.out.println("My Runnable");

}};

Lambda Expression : lambda expressions are means to create anonymous classes of functional interfaces easily. There are no runtime benefits of using lambda expressions.

So use it carefully.

Some of the useful java 8 functional interfaces are Consumer, Supplier, Function and Predicate.

Lambda Expression are the way through which we can visualize **functional programming** in the java object oriented world.

Java language provide support for using lambda expressions only with functional interfaces.

**Why do we need Lambda Expression?**

1. Reduced lines of codes: one of the best example is functional interface rather than using anonymous java class
2. Sequential and Parallel Execution Support: using lambda expression we can benefit from STREAM API sequential and parallel operation support.
3. Passing behaviours into methods.
4. Higher efficiency with laziness.

**Operators in Java 8**

:: double colon operator, it is used for the method reference its short form of lambda expression is i-> method name.

**Lambda Expression Example:**

() -> {} //no parameters, void result

() -> 42 // no parameters, expression body

() ->{return 42;}// no parameters, block body with return.

//complex block body with multiple return

() -> { if (true) return 10; else { int result = 15; for (int i = 1; i < 10; i++) result \*= i; return result; } }

(int x) -> x+1;//Single declared type argument.

(x) -> x+1 //Single inferred type argument

X -> x+1 //Parenthesis optional for single inferred-type case.

**Method and constructor references**

A method reference is used to refer to a method without invoking it.

A constructor reference is similarly used to refer to a constructor without creating a new instance of the named class or array type.

System::getProperty //method reference

System.out::println //method refrence

“abc”::length //method reference

ArrayList::new //method reference

Int[]::new

**Lambda expression tutorial**

<https://www.journaldev.com/2763/java-8-functional-interfaces>

**Functional Programming vs Object Oriented Programming:**

If we look into some other programming languages such as C++, JavaScript; they are called **functional programming language** because we can write functions and use them when required.

Some of these languages support Object Oriented Programming as well as Functional Programming.

Java 7 updates

String in switch statement.

Binary Literal.

The try-with-resources or ARM (Automatic resource management).

Catching multiple exceptions by single catch.

Underscores in numeric literals.

**Multithreading interview questions**

<https://www.journaldev.com/1162/java-multithreading-concurrency-interview-questions-answers>

<https://www.educba.com/java-multi-threading-interview-questions/>

* Many other java threads running in background like memory management, system management, signal processing etc.
* Callable interface uses Generics, so it can return any type of Object.
* *Runnable* tasks can be run using the *Thread* class or *ExecutorService* whereas*Callables* can be run only using the latter.
* *Runnable*is the core interface provided for representing multi-threaded tasks and *Callable* is an improved version of *Runnable* that was added in Java 1.5.

what is mean by the production quality code?

Handling invalid input and boundary condition called production quality code.

Arithmetic VS logical right shift operators?

In a logical right shift operator, we shift the bits and put a 0 in the most significant bit.it is indicated with the >>> operator.

In an arithmetic right shift, we shift values to the right but fill in the new bits with the value of the sign bit. This has the effect of (roughly) dividing by two.it is indicated by a >> operator.

* To perform binary search on the given array/List it must be on sorted array. Time complexity of the liner search algorithm is O (n) while binary search have O(log n) binary search is a faster.
* In a class two overloaded methods are there one takes String as input argument and second takes Object as input argument. When we call method by pass input as “Test” which method will call? Or in a class two public static void main method is there one takes String array and another takes Object array is valid or not ?
* Questions on Enum: enum introduce in the java 1.5, we can use it to define the constants

<http://www.java67.com/2013/07/15-java-enum-interview-questions-amswers-for-experienced-programmers.html>

<http://www.java67.com/2012/08/string-to-enum-in-java-conversion.html>

* When concurrent modification exception will occur and how to avoid it: Refer Technical Key points Sheet—Exception section to see the details .Checked exception is compile time exceptip l 7.o jko,l0pok’/./ mkon and unchecked exception is runtime exception.
* In a tree set we are inserting the data 100 int number and in loop after each insert calling remove (i-1) then what will be the output?: It will run properly and give the output of last element not all elements which got inserted into the set. remove(i-1) will keep on removing the previous element index place and will keep only the current index value. So after the;. loop completion the current value will be last index value i.e 99.ol

Tagarem interview Question:

1. “= =” and equal method what is the difference?

Equ ality operator is a binary operator and use to compare the primitive and object type both.

“= =” compare two objects based on the memory refrence. So “= =” operator will return true only if only if two object reference it is comparing represent exactly same object otherwise "==" will return false.

In case of String comparison == will return true if both the references are pointing to same object. But equal method will compare the content of the String.

**Summary**

1) use == to compare primitive e.g. boolean, int, char etc, while use equals() to compare objects in Java.

2) == return true if two reference are of same object. Result of equals() method depends on overridden implementation.

3) For comparing String use equals() instead of  == equality operator.

Read more: <https://javarevisited.blogspot.com/2012/12/difference-between-equals-method-and-equality-operator-java.html#ixzz5mVeXDw7k>   
  
Read more: <https://javarevisited.blogspot.com/2012/12/difference-between-equals-method-and-equality-operator-java.html#ixzz5mVcnzJtX>

1. If two application share same JVM and one string lateral defined in the one application can second used it?
2. If str=”Hello”, and String str1=new String(“Hello”); what will be the output of the below str==str1 and str.equal(str1)
3. What will be the output of the below?

String str2=new String(new String(“Hello”));

1. Sort a student first name is descending order and if first name are equal then sort based on the last name?
2. We have an arr={1,0,2,3,4,5,0,0,6}; modify array such that all non-zero should come first and zero will come last. “Move all zero end of the array.”
3. What is the difference between bean and java POJO class.?

All JavaBeans are **POJOs** but not all **POJOs** are JavaBeans.

A JavaBean is a Java object that satisfies certain programming conventions:

The JavaBean class must implement either Serializable or Externalizable.

The JavaBean class must have a public no-arg constructor.

All JavaBean properties must have public setter and getter methods (as appropriate).

All JavaBean instance variables should be private

pojo class is an ordinary class without any specialties,class totally loosely coupled from technology/framework.the class does not implements from technology/framework and does not extends from technology/framework api that class is called pojo class.

Microservices:

<https://www.edureka.co/blog/what-is-microservices/>