Alok Sarkar

[Company name]  [Company address]

JAVA - CORE

**Explain Marshalling and unmarshalling.**

**Answer.** **Marshalling:** When a client invokes a method that accepts parameters on a remote object, it bundles the parameters into a message before sending it over the network. These parameters can be of primitive type or objects. When the parameters are of primitive type, they are put together and a header is attached to it. If the parameters are objects, then they are serialized. This process is called marshalling.

**Unmarshalling:** The packed parameters are unbundled at the server-side, and then the required method is invoked. This process is called unmarshalling.

**What is Double Brace initialization in Java?**

**Answer.** Double brace initialization in Java is a combination of two separate Java processes. When we use the initialization block for an anonymous inner class it becomes double brace initialization in Java. The inner class that we created will have a reference to the enclosing outer class. We can use that reference using the ‘this’ pointer.

**What is the difference between a ClassNotFoundException and NoClassDefFoundError?**

**Answer.** ClassNotFoundException and NoClassDefFoundError exceptions occur when a particular class is not found during the runtime. However, they differ from each other and occur in different scenarios.

A ClassNotFoundException is an exception that occurs when we try to load a class during the runtime using methods like Class.forName() or loadClass() methods and these classes are not found in the classpath. Whereas NoClassDefFoundError is an error that occurs when a particular class is present at compile-time but missing at run time.

|  |  |
| --- | --- |
| **ClassNotFoundException** | **NoClassDefFoundError** |
| It is an exception. It is of type java.lang.Exception. | It is an error. It is of type java.lang.Error. |
| It occurs when an application tries to load a class at runtime which is not present in the classpath. | It occurs when the Java runtime system does not find a class definition, which is present at compile-time but missing at run time. |
| It is thrown by methods like ,loadClass(), Class.forName(), and findSystemClass(). | Java Runtime System throws this error. |
| It occurs when there is no update of classpath with required JAR files. | It occurs when the definition of the required class is missing at runtime. |

**What is the difference between final, finalize, and finally?**

**Answer.** Below is a list of differences between final, finally and finalize:

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **final** | **finally** | **finalize** |
| 1) | The final keyword applies restrictions on class, method, and variable. We can not inherit the final class, we cannot change the value of a final variable and also can not override the final method. | The finally block places the important code. This code will be executed whether an exception is handled or not. | The finalize method performs the cleanup processing just before the object is garbage collected. |
| 2) | The Final is a keyword. | Finally is a block. | Finalize is a method. |

**What is a Server Side Include (SSI)?**

**Answer.** Server Side Includes (SSI) is a simple and interpreted server-side scripting language. SSI is used almost exclusively for the Web. It is embedded with a servlet tag. Including the contents of one or more than one file into a Web page on a Web server is the most frequent use of SSI. When a browser accesses a Web page, the Web server replaces the servlet tag on that Web page with the hypertext generated by the corresponding servlet.

**How can you find out what client machine is making a request to your servlet ?**

**Answer.** There is a ServletRequest class that has functions for finding out the IP address or hostname of the client machine. The getRemoteAddr() method gets the IP address of the client machine and getRemoteHost() method gets the hostname of the client machine.

**What is HTTP Tunneling?**

**Answer.** HTTP Tunneling is a mechanism that encapsulates the communications performed using various networks using the HTTP or HTTPS protocols. Therefore, the HTTP protocol acts as a wrapper for a channel that the network protocol being tunneled uses to communicate. HTTP Tunneling is the masking of other protocol requests as HTTP requests.

**What is the difference between creating the String as a literal and with a new operator?**

**Answer.** When we create an object of String in Java using a new() operator, it is created in a heap memory area and not into the String pool. But when we create a String using literal, then it gets stored in the String pool itself. The String pool exists in the PermGen area of heap memory.

For example,

String str = **new** String("java");

The above statement does not put the String object str in the String pool. We need to call the String.intern() method to put the String objects into the String pool explicitly.

It is only possible when we create a String object as String literal.

For example,

String str1 = "java";

Java automatically puts the String object into the String pool.

**Why is the Char array preferred over String for storing passwords?**

**Answer.** As we know that String is immutable in Java and stored in the String pool. Once we create a String, it stays in the String pool until it is garbage collected. So, even though we are done with the password it is still available in memory for a longer duration. Therefore, there is no way to avoid it.

It is clearly a security risk because anyone having access to a memory dump can find the password as clear text. Therefore, it is preferred to store the password using the char array rather than String in Java.

**What is the difference between creating an object using new operator and Class.forName().newInstance()?**

**Answer.** The new operator statically creates an instance of an object. Whereas, the newInstance() method creates an object dynamically. While both the methods of creating objects effectively do the same thing, we should use the new operator instead of Class.forName(‘class’).getInstance().

The getInstance() method uses the Reflection API of Java to lookup the class at runtime. But, when we use the new operator, Java Virtual Machine will know beforehand that we need to use that class and therefore it is more efficient.

Can you have virtual functions in Java?

Yes, all functions in Java are virtual by default.

### What is the final blank variable?

A final variable, not initialized at the time of declaration, is known as the final blank variable. We can't initialize the final blank variable directly. Instead, we have to initialize it by using the class constructor. It is useful in the case when the user has some data which must not be changed by others

Q: Why do we need marker interface ?

A: check at runtime, what is the object instance.

Q: what is serialization? Why do we need it?

Q: MapStruct vs Dogger?

Q: Mapstruct vs ModelMapper?

Q: Mapstruct vs BeanUtils?

Q: New features from Java 8 to 21?

Q Create your own hash-map?

Q: Create your own immutable class?

Q: HashMap vs HashTable? Which is better?

Q: what is concurrent HashMap?

Q: Exception Hierarchy?

Q: what is SOLID design principles?

Q: inner class vs nested class?

Q: Java keycerts vs keystores?

Q: filters vs interceptors?

Q: What is Class Loader in Java.