Topic Name 🡪 Input output stream Serialization and Deserialization Array and Properties file String .

Q1) difference between string, string buffer and string builder ?

|  |  |
| --- | --- |
| **StringBuffer** | **StringBuilder** |
| StringBuffer is synchronized i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. | StringBuilder is non-synchronized i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. |
| StringBuffer is less efficient than StringBuilder. | StringBuilder is more efficient than StringBuffer. |
| StringBuffer was introduced in Java 1.0 | StringBuilder was introduced in Java 1.5 |

Q2) write a code to calculate 2nd Highest no by using array ?

Select salary from(select salary from student ORDER by salary DESC LIMIT2)As s ORDER by salary LIMIT1;

We have first sorted all salaries from employee table in descending order so that 2nd highest salaries comes at top of the result set.

Q3) Transient use ?

The transient keyword can be used with the data members of a class in order to avoid their serialization.

For example, if a program accepts a user's login details and password. But we don't want to store the original password in the file.

Here, we can use transient keyword and when JVM reads the transient keyword it ignores the original value of the object and instead stores the default value of the object.

Syntax-

**private** **transient** <member variable>;

Q4) Difference between final and immutable ?

* The Final in Java is a keyword that is relevant only to variables, methods, and classes.
* The Java Final Keyword is used to restrict the user to use variables, methods, and classes.
* The Final keyword is applicable for all the types of variables like instance variable, static variable, and local variable.
* Immutability means once we create an object it is not permissible to modify the content of that object.
* If any person trying to change the content if there is a change in the content with those changes a new object will be generated.
* If there are no changes in the content existing object will be reused.

Q5)Can we write final keywords with string.

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.

Q6) write 2-dimensional array syntax ?

Q7) What is externalisation?

Externalization in Java is used to customize the serialization mechanism.

Java serialization is not much efficient.

When we have bloated objects that hold several attributes and properties,

it is not good to serialize them. Here, the externalization will be more efficient.

|  |  |  |
| --- | --- | --- |
| **Key** | **Serialization** | **Externalization** |
| Interface | It is a marker interface | Whereas, it contains two methods readExternal() and writeExternal() |
| Process | Default serialization | Custom serialization |
| UID | serialVersionUID | No UID |
| Storage | Stores the data that have objects | Directly stores objects |
| Access | No access | Complete control to customize the serialization |
| Performance | It provides relatively slow performance. | It provides full control over the implementation approach. |

Q8) Why string is immutable ?

In Java, **String objects are immutable.**

Immutable simply means unmodifiable or unchangeable.

Once String object is created its data or state can't be changed but a new String object is created.

Q9) Diff string literal & heap area ?

**Heap Memory**

* The heap memory is a run time data area from which the memory for all java class instances and arrays is allocated.
* The heap is created when the JVM starts up and may increase or decrease in size while the application runs.
* The size of the heap can be specified using –Xms VM option. The heap can be of fixed size or variable size depending on the garbage collection strategy. Maximum heap size can be set using –Xmx option.
* By default, the maximum heap size is set to 64 MB.

**String Constant Pool**

* String uses a special memory location to reuse of String objects called String Constant Pool.
* String objects created without the use of new keyword are stored in the String Constant Pool part of the heap.
* One of the important characteristics of String constant pool is that it does not create the same String object if there is already String constant in the pool.

Q10) "string objects means pool and heap concepts, how much objects will be created from 1) String s=”amit” 2) String s=”amit” 3) String s=new string ”amit”" String code.

Q11) intern method ?

The **Java String class intern()** method returns the interned string. It returns the canonical representation of string.

It can be used to return string from memory if it is created by a new keyword. It creates an exact copy of the heap string object in the String Constant Pool.

Why do we need intern method in Java?

Java String intern() method is used for getting the string from the memory if it is already present. This method ensures that all the same strings share the same memory.

Q12) **Cloneable** and serializable ?

An [interface](https://www.javatpoint.com/interface-in-java) that does not contain methods, fields, and constants is known as **marker interface.**

The **Serializable** and **Cloneable** interfaces are the example of marker interface.

**Serializable🡪**

It is a marker interface in Java that is defined in the**java.io** package.

If we want to make the class serializable, we must implement the **Serializable** interface.

If a class implements the Serializable interface, we can serialize or deserialize the state of an object of that class.

**Cloneable🡪**

**Cleanable interface** in Java is also a marker interface that belong to **java.lang** package.

It generates replica (copy) of an object with different name.

We can implement the interface in the class of which class object to be cloned. It indicates the **clone()** method of the Object class.

If we do not implement the Cloneable interface in the class and invokes the clone() method, it throws the **ClassNotSupportedException.**

Q13) Serial version uid ?

SerialVersionUID is a unique identifier for each class, JVM uses it to compare the versions of the class ensuring that the same class was used during Serialization is loaded during Deserialization.

Q14) String buffer is thread safe or string is thread safe. ?

* StringBuffer is thread-safe meaning that they have synchronized methods to control access so that only one thread can access StringBuffer object's synchronized code at a time.
* StringBuffer objects are generally safe to use in a multi-threaded environment where multiple threads may be trying to access the same StringBuffer object at the same time.
* A string is immutable and therefore thread safe.

Serialization Method

|  |  |
| --- | --- |
| **Method** | **Description** |
| 1) public final void writeObject(Object obj) throws IOException {} | It writes the specified object to the ObjectOutputStream. |
| 2) public void flush() throws IOException {} | It flushes the current output stream. |
| 3) public void close() throws IOException {} | It closes the current output stream. |