

1. What are the difference between equals() and "=="?

Ans.

The difference between equals() and "==" are given below -

== (Operator)	equals() (Method)
i) Used for comparing primitive types (like int, boolean, char) and object references.	i) Used exclusively for comparing objects.
ii) For primitive types, it compares their actual values. For objects, it compares their memory addresses (references), checking if two references point to the exact same object in memory.	ii) Compares the content or state (value) of two objects.
iii) Cannot be overridden.	iii) Can be overridden.

2. Java string are immutable, why?

Ans.

In Java, string are immutable, meaning once a String object is created, its value cannot be changed. This design choice was made for several important reasons:

i) Security:

- Strings are widely used to store sensitive information (like usernames, passwords, file paths, network connections).
- If String objects were mutable, malicious code could alter their contents after creation.
- Immutability ensures that sensitive data cannot be changed unexpectedly.

ii) String Pooling and Performance:

- Java maintains a String pool to reduce memory usage.
- Since strings are immutable, Java can safely reuse instances.

iii) Thread Safety:

- Immutability means multiple threads can access the same string object without synchronization.
- This makes strings inherently thread-safe.

iv) Caching Hash Codes:

- String objects are often used as keys in hash-based collections (like HashMap).

v) Predictable Behavior:

- With immutability, once a String is created, it behaves consistently throughout the program.

Java Strings are immutable for security, efficiency, thread-safety, and reliability. This design helps ensure better performance and consistent behavior in Java programs.