ARNOB SEN
2938520103

What is the difference between (a==b) and a.equals(b)

Ans:

In Java programming, the distinction between a.equals(b) and a==b lies in the type of comparison performed:

a==b (Equality operator):

primitive Types: For primitive data types (like int, char, float, boolean), == compares the actual values stored in a and b.

Object Types: For object types, == compares the references or memory addresses of the objects. It does not compare the content or state of the objects.

a.equals(b) (Equals Method):

Primitive Types: The equals() method cannot be directly used with primitive types as they are not objects and do not have methods.

Object Types: The equals method is defined in the object class and can be overridden by subclasses. By default (in the object class), equals() behaves similarly to == and compares object refere

Why java strings are immutable?

Ans:

Java strings are immutable, meaning their content cannot be changed once created. This design choice provides several significant benefits.:

Security:

Strings are often hold sensitive information like password, usernames, or file paths. Immutability ensures that once these value are set and potentially validated, they cannot be altered by malicious code or accidental modification.

Thread Safety:

Immutable objects are inherently thread-safe because their state cannot be changed.

String Pooling and Memory Efficiency:

Java utilizes a "string pool" where literal strings objects are stored.

Hash code Caching:

Strings are frequently used as keys in hash-based collection like HashMap or HashSet.