Task 1: List vs Vector

Thread Safety: List is not thread-safe by default. Vector is synchronized, making it thread-safe.

Performance: List is generally faster in single-threaded environments due to no synchronization overhead. Vector can be slower because of synchronization.

Synchronization: List implementations (like ArrayList) are not synchronized, while Vector has synchronized methods.

Task 2: HashSet vs LinkedHashSet

Order of Elements: HashSet does not maintain insertion order. LinkedHashSet maintains insertion order.

Performance: Both have similar performance for basic operations, but LinkedHashSet may have slightly more overhead due to maintaining order.

Use Cases: Use HashSet when you only care about uniqueness. Use LinkedHashSet when you also need to preserve insertion order.

Task 3: String vs StringBuilder vs StringBuffer

Mutability: String is immutable. StringBuilder and StringBuffer are mutable.

Thread Safety: String is inherently thread-safe due to immutability. StringBuffer is synchronized and thread-safe. StringBuilder is not thread-safe.

Performance: StringBuilder is faster than StringBuffer because it does not have synchronization overhead. String operations create new objects, which is less efficient for modifications.

When to Use: Use String when the value will not change. Use StringBuilder in single-threaded contexts when modifying strings. Use StringBuffer in multi-threaded contexts where string modifications are needed.