10 Scenario-Based Collection Decisions

1.]
Scenario: Maintain order of attendance list
Collection: LinkedHashSet<>
Reason: Does not allow duplicate values therefore no student ID is entered twice even my mistake + maintains insertion order therefore goal is achieved.

2.]
Scenario: Remove duplicates from exam submissions
Collection: HashSet<>
Reason: Exam submissions require ID, ID's submitted multiple times will get removed except the first time it was submitted + HashSet<> uses hashing making the process fast and efficient.
In-Short: Removes duplicates + fast and efficient.

3.]
Scenario: College timetable -> sorted by time
Collection: TreeMap<>
Reason: College time table consists of time associated with subject therefore Map<> is used + TreeMap<> sorts the keys automatically in ascending order.

4.]
Scenario: Student roll number & name lookup
Collection: HashMap<>
Reason: Student roll number as key and student name as value + fast and efficient to get value associated with the student roll number.

5.]
Scenario: Manage print queue jobs
Collection: LinkedList<>
Reason: Dynamic and fast insertion (enqueue(x)) & removal (dequeue()) + Designed for queue like operations.

6.]
Scenario: Keep browser back–forward history
Collection: LinkedList<>
Reason: Maintains visiting order of websites + allows duplicate entries of website in the history + allows forward and backward traversal.

7.]
Scenario: Store leaderboard scores sorted
Collection: TreeSet<>
Reason: TreeSet<> : If scores cannot be tied + score and team name / player name + sorts team name / player name according to the score

---

8.]
Scenario: Store products in ecommerce
Collection: HashMap<>
Reason: Product ID for key and product description for value + No duplicate product IDs allowed + fast and efficient retrival of product.

9.]
Scenario: Cache with insertion order (Assuming cache data consists of duplication and data is not in <key, value> format)
Collection: LinkedList<>
Reason: Allows duplication of values + maintains insertion order of data.

10.]
Scenario: Manage tasks by priority
Collection: PriorityQueue<Task>
Reason: We can set priority of tasks + execute according to priority

---