

 **Java Set Interface Programming Question Bank (Scratch to Advanced)** **BASIC LEVEL QUESTIONS**

1. Create a HashSet and add 5 student names.
 2. Check if a specific element exists in a HashSet.
 3. Iterate over a HashSet using iterator and for-each loop.
 4. Remove a specific element from a HashSet.
 5. Clear a HashSet.
 6. Check if the HashSet is empty.
 7. Convert a HashSet to an array.
 8. Create a TreeSet to store 10 integers and display them in sorted order.
 9. Create a LinkedHashSet to preserve insertion order of cities.
 10. Clone a HashSet.
 11. Compare two sets and retain common elements.
 12. Create a union of two HashSets.
 13. Create an intersection of two HashSets.
 14. Remove all elements of one set from another.
 15. Convert an array to a HashSet.
-

 **INTERMEDIATE LEVEL QUESTIONS**

16. Sort a HashSet using TreeSet.
17. Count unique characters in a string using Set.
18. Remove duplicate characters from a string using Set.
19. Find the first duplicate word in a sentence.
20. Check if two sets are equal.
21. Remove duplicate integers from a List using HashSet.

-
22. Convert a Set to List and vice versa.
 23. Difference between HashSet, LinkedHashSet, and TreeSet.
 24. Merge two sets and find total unique elements.
 25. Get the size of a Set after multiple operations.
-

ADVANCED / SCENARIO-BASED QUESTIONS

26. Remove duplicates from a list of employees based on id using Set.
 27. Detect duplicate entries in a registration system using Set.
 28. Group email addresses and print only unique domain names.
 29. Use TreeSet with a custom comparator to sort strings by length.
 30. Maintain insertion order while ensuring uniqueness using LinkedHashSet.
 31. Validate unique coupon codes entered by user.
 32. Find all duplicate words in a paragraph.
 33. Validate unique roll numbers for student entries.
 34. Implement a lottery system picking unique winners from entries.
 35. Identify common elements between two product lists from different vendors.
-

COMPANY-ASKED INTERVIEW QUESTIONS

36. **TCS** – Remove duplicates from an array without using collection.
37. **Capgemini** – Find the first non-repeating character in a string using Set and Map.
38. **Infosys** – Identify duplicate numbers in an array using Set.
39. **Wipro** – Find union and intersection of two arrays using Set.
40. **Accenture** – Write a function to check if all characters in a string are unique.
41. **Cognizant** – Create a system to track unique user logins.
42. **Tech Mahindra** – Validate uniqueness of usernames in registration form.
43. **Persistent** – Count the frequency of each word and remove duplicates from a sentence.

44. **L&T Infotech** – Prevent uploading of duplicate files using a Set.
45. **Hexaware** – Store only unique values from a specific column in a CSV.