Experiment No: 11

AIM: To demonstrate the use of Collection framework.

Date:

CO mapped: CO-4

Objectives:

a) To proficiently demonstrate the use of Java's Collection framework, including understanding its core interfaces (List, Set, Map), implementing and manipulating data structures like lists, sets, and maps, and effectively applying collections for data storage, retrieval, and manipulation in Java applications.

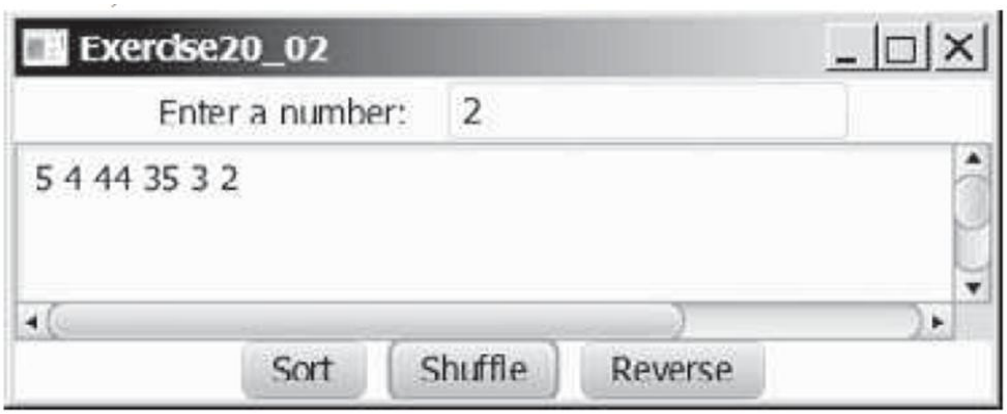
b) Mastery of the Java Collection framework is essential for managing and organizing data efficiently in Java applications. This objective focuses on understanding the core collection interfaces and using them to build versatile data structures to meet various application needs.

Background:

The Collection in Java is a framework that provides architecture to store and manipulate a group of objects. Java Collections can achieve all the operations that you perform on data such as searching, sorting, insertion, manipulation, and deletion. Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

Practical questions:

1. Write a program that lets the user enter numbers from a graphical user interface and displays them in a text area, as shown in Figure. Use a linked list to store the numbers. Do not store duplicate numbers. Add the buttons Sort, Shuffle, and Reverse to sort, shuffle, and reverse the list.



2. Create two priority queues, {"George", "Jim", "John", "Blake", "Kevin", "Michael"}

and {"George", "Katie", "Kevin", "Michelle", "Ryan"}, and find their union, difference, and intersection.

3. Store pairs of 10 states and its capital in a map. Your program should prompt the user to enter a state and should display the capital for the state.

Observations: Put Output of the program

Conclusion:

Quiz:

1. Write a note on 'Collection in JAVA'. Also discuss List and Enumeration Interface.
2. Differentiate between Enumeration and Iterator.
3. Compare List, Set and Map interfaces. Also compare ArrayList, TreeSet and HashMap classes in java.
4. Explain the unique features of Map interface.
5. How do you perform common operations like sorting, searching, or filtering on Collections?

Suggested Reference:

1. https://www.tutorialspoint.com/java/
2. https://www.geeksforgeeks.org/
3. https://www.w3schools.com/java/
4. https://www.javatpoint.com/