**ArrayList and LinkedList:**

1. Create an ArrayList to store the names of students in a class. Add, remove, and print the list of students.
2. Write a program to insert elements into the linked list at the first and last positions. Also check if the linked list is empty or not.

**Iteration and Operations:**

1. Write a program that iterates through a List of numbers and prints only the even numbers.
2. Write a program to Implement a method to shuffle the elements of an ArrayList.
3. Write a method to reverse the order of elements in a LinkedList.

**Common Methods:**

1. Create a program that uses addAll() to combine two ArrayLists into one. Print the merged list.
2. Use the subList() method to extract a portion of a List .

**Stack**

1. Create a Stack and perform the following operations:

* Push elements "A", "B", and "C" onto the stack.
* Pop an element from the stack.
* Push elements "D" and "E" onto the stack.
* Peek at the top element without removing it.
* Print the stack.

**Queue**

1. Initialize a queue and perform the following operations:

* Enqueue elements "A", "B", and "C" into the queue.
* remove an element, “A” from the queue.
* Enqueue elements "D" and "E" into the queue.
* Peek at the front element without removing it.
* Print the elements in Queue.

**Set Operations**

1. Write a program to create a HashSet to store unique integers. Check and handle the case if an element already exists before adding.
2. Write a program utilizing a HashSet to showcase fundamental Set operations: creating a HashSet, adding various elements (e.g., strings, integers), printing the set's elements, checking the existence of a specific element, removing an element, clearing all elements, and finally, printing the resultant state of the set.
3. Implement a TreeSet to store unique names in alphabetical order.
4. Write a Java program to create a tree set, add some colors (strings) and print out the tree set.

**Map(HashMap, LinkedHashMap, TreeMap):**

1. Write a program that uses a HashMap to store contact information (name and phone number).
2. Use a TreeMap to sort a map of products based on their prices.
3. Check if a specific key exists in a HashMap. If it does, update the corresponding value.
4. Create a LinkedHashMap to store the order in which books were added to a library.

**Collection Algorithm**

**Sorting**

1. Write a program that sorts an array of integers using the sort() method. Also try sorting in reverse order.
2. Write a program that sorts an array list of string of colors using the sort() method. Also try sorting in reverse order.

**Binary search**

1. Write a program to initialize an ArrayList with a set of integers. Implement a binary search algorithm to find a particular integer.