

# Assignment 7

**Q1. Write a java program to perform the following basic ArrayList operations:**

- i. Add all elements to ArrayList.**
- ii. Read all elements in ArrayList by using an iterator.**
- iii. Check whether ArrayList contains some specific elements or not.**
- iv. Sort the ArrayList contents.**
- v. Swap two elements of ArrayList.**
- vi. Get sublist from ArrayList and print it**
- vii. Add a ArrayList1 to another ArrayList2**
- viii. Retain ArrayList1 elements**
- ix. Remove all elements from ArrayList**

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;

public class ArrayListOperations {

    public static void main(String[] args) {

        ArrayList<String> arrayList = new ArrayList<>();

        arrayList.add("Apple");

        arrayList.add("Banana");

        arrayList.add("Cherry");

        arrayList.add("Date");

        arrayList.add("Elderberry");

        arrayList.add("Fig");

        arrayList.add("Grape");

        System.out.println("Elements in the ArrayList:");

        Iterator<String> iterator = arrayList.iterator();

        while (iterator.hasNext()) {

            System.out.println(iterator.next());
```

```

}

String searchElement = "Cherry";

if (arrayList.contains(searchElement)) {

    System.out.println("ArrayList contains: " + searchElement);

} else {

    System.out.println("ArrayList does not contain: " + searchElement);

}

Collections.sort(arrayList);

System.out.println("Sorted ArrayList: " + arrayList);

int index1 = 1; // Banana

int index2 = 3; // Date

System.out.println("Before swapping: " + arrayList);

Collections.swap(arrayList, index1, index2);

System.out.println("After swapping index " + index1 + " and index " + index2 + ": " + arrayList);

List<String> subList = arrayList.subList(2, 5); // From index 2 to 4

System.out.println("Sublist from index 2 to 4: " + subList);

ArrayList<String> arrayList2 = new ArrayList<>();

arrayList2.add("Honeydew");

arrayList2.add("Kiwi");

arrayList2.add("Lemon");

arrayList2.addAll(arrayList);

System.out.println("ArrayList2 after adding ArrayList1: " + arrayList2);

arrayList2.retainAll(arrayList);

System.out.println("ArrayList2 after retaining elements from ArrayList1: " + arrayList2);

arrayList.clear();

System.out.println("ArrayList after removing all elements: " + arrayList);

}

}

```

**Q2. Write a java program to perform the following operations on Vector:**

**i. Add all elements to Vector and print**

**ii. Check the first and last occurrence of duplicate elements in vector**

```
import java.util.Vector;

public class VectorOperations {

    public static void main(String[] args) {

        Vector<String> vector = new Vector<>();

        vector.add("Apple");

        vector.add("Banana");

        vector.add("Cherry");

        vector.add("Date");

        vector.add("Elderberry");

        vector.add("Fig");

        vector.add("Grape");

        System.out.println("Elements in the Vector:");

        for (String element : vector) {

            System.out.println(element);

        }

        String[] elementsToCheck = {"Banana", "Cherry"};

        for (String element : elementsToCheck) {

            int firstIndex = vector.indexOf(element);

            int lastIndex = vector.lastIndexOf(element);

            if (firstIndex != -1) {

                System.out.println("Element " + element + " - First occurrence: " + firstIndex + ", Last occurrence: " + lastIndex);

            } else {

                System.out.println("Element " + element + " not found in the vector.");

            }

        }

    }

}
```

```
}
```

**Q3. Write a java program to perform the following operations on Stack:**

**i. Push elements on stack**

**ii. Print element on top of stack**

**iii. Pop all elements from stack one by one**

```
import java.util.Stack;

public class StackOperations {

    public static void main(String[] args) {

        Stack<String> stack = new Stack<>();

        stack.push("Apple");
        stack.push("Banana");
        stack.push("Cherry");
        stack.push("Date");
        stack.push("Elderberry");

        if (!stack.isEmpty()) {

            System.out.println("Element on top of the stack: " + stack.peek());

        } else {

            System.out.println("The stack is empty.");

        }

        System.out.println("Popping elements from the stack:");

        while (!stack.isEmpty()) {

            System.out.println("Popped: " + stack.pop());

        }

        if (stack.isEmpty()) {

            System.out.println("The stack is now empty.");

        }

    }

}
```

**Q4. Write a java program to perform the following HashSet operations:**

- i. Add the elements into HashSet.**
- ii. Iterate through HashSet using iterator, using forLoop and print the elements.**
- iii. Copy Set content to another HashSet.**
- iv. Delete all elements from HashSet.**
- v. Copy all elements from HashSet to an array.**
- vi. Eliminate duplicate user defined objects from HashSet.**
- vii. Search an element in a HashSet.**

```
import java.util.HashSet;
import java.util.Iterator;
class Student {
    String name;
    int id;
    Student(String name, int id) {
        this.name = name;
        this.id = id;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (!(obj instanceof Student)) return false;
        Student student = (Student) obj;
        return id == student.id && name.equals(student.name);
    }
    @Override
    public int hashCode() {
        return 31 * name.hashCode() + id;
    }
    @Override
```

```

    public String toString() {
        return "Student{name='" + name + "', id='" + id + "'}";
    }
}

public class HashSetOperations {
    public static void main(String[] args) {
        HashSet<String> hashSet = new HashSet<>();
        hashSet.add("Apple");
        hashSet.add("Banana");
        hashSet.add("Cherry");
        hashSet.add("Date");
        System.out.println("Iterating through HashSet using Iterator:");
        Iterator<String> iterator = hashSet.iterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }
        System.out.println("\nIterating through HashSet using for-loop:");
        for (String fruit : hashSet) {
            System.out.println(fruit);
        }
        HashSet<String> anotherHashSet = new HashSet<>(hashSet);
        System.out.println("\nCoped HashSet: " + anotherHashSet);
        hashSet.clear();
        System.out.println("\nHashSet after deleting all elements: " + hashSet);
        String[] array = anotherHashSet.toArray(new String[0]);
        System.out.println("\nElements copied to array:");
        for (String fruit : array) {
            System.out.println(fruit);
        }
        HashSet<Student> studentHashSet = new HashSet<>();
        studentHashSet.add(new Student("John Doe", 1));
    }
}

```

```
studentHashSet.add(new Student("Jane Smith", 2));
studentHashSet.add(new Student("John Doe", 1)); // Duplicate
System.out.println("\nStudent HashSet (duplicates eliminated):");
for (Student student : studentHashSet) {
    System.out.println(student);
}
String searchElement = "Cherry";
if (anotherHashSet.contains(searchElement)) {
    System.out.println("\nElement " + searchElement + " found in HashSet.");
} else {
    System.out.println("\nElement " + searchElement + " not found in HashSet.");
}
}
}
```

**Q5. Perform the following Mathematical Operations on set:**

**i. Union**

**ii. Intersection**

**iii. Difference**

**iv. Symmetric Difference**

```
import java.util.HashSet;

public class SetOperations {

    public static void main(String[] args) {

        HashSet<Integer> setA = new HashSet<>();

        HashSet<Integer> setB = new HashSet<>();

        setA.add(1);
        setA.add(2);
        setA.add(3);
        setA.add(4);
        setA.add(5);

        setB.add(4);
        setB.add(5);
        setB.add(6);
        setB.add(7);
        setB.add(8);

        HashSet<Integer> unionSet = new HashSet<>(setA);
        unionSet.addAll(setB);

        System.out.println("Union of Set A and Set B: " + unionSet);

        HashSet<Integer> intersectionSet = new HashSet<>(setA);
        intersectionSet.retainAll(setB);

        System.out.println("Intersection of Set A and Set B: " + intersectionSet);

        HashSet<Integer> differenceSet = new HashSet<>(setA);
        differenceSet.removeAll(setB);

        System.out.println("Difference of Set A and Set B (A - B): " + differenceSet);
```



```
HashSet<Integer> symmetricDifferenceSet = new HashSet<>(setA);
symmetricDifferenceSet.addAll(setB);

HashSet<Integer> intersectionForSymmetric = new HashSet<>(setA);
intersectionForSymmetric.retainAll(setB);

symmetricDifferenceSet.removeAll(intersectionForSymmetric);

System.out.println("Symmetric Difference of Set A and Set B: " + symmetricDifferenceSet);
}
}
```

**Q6. Write a java program to perform the following basic HashMap operations:**

**i. Add and Iterate through HashMap.**

**ii. Search a key in HashMap.**

**iii. Search a value in HashMap.**

**iv. Get all keys from HashMap.**

**v. Get all values from HashMap**

**vi. Delete all elements from HashMap.**

```
import java.util.HashMap;
import java.util.Map;
public class HashMapOperations {
    public static void main(String[] args) {
        HashMap<Integer, String> hashMap = new HashMap<>();
        hashMap.put(1, "Apple");
        hashMap.put(2, "Banana");
        hashMap.put(3, "Cherry");
        hashMap.put(4, "Date");
        hashMap.put(5, "Elderberry");
        System.out.println("Iterating through HashMap:");
        for (Map.Entry<Integer, String> entry : hashMap.entrySet()) {
            System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());
        }
        int searchKey = 3;
        if (hashMap.containsKey(searchKey)) {
            System.out.println("\nKey " + searchKey + " found with value: " + hashMap.get(searchKey));
        } else {
            System.out.println("\nKey " + searchKey + " not found.");
        }
        String searchValue = "Banana";
        if (hashMap.containsValue(searchValue)) {
            System.out.println("Value '" + searchValue + "' found.");
        }
    }
}
```

```
} else {  
    System.out.println("Value "" + searchValue + "" not found.");  
}  
  
System.out.println("\nAll keys in the HashMap:");  
for (Integer key : hashMap.keySet()) {  
    System.out.println(key);  
}  
  
System.out.println("\nAll values in the HashMap:");  
for (String value : hashMap.values()) {  
    System.out.println(value);  
}  
  
hashMap.clear();  
  
System.out.println("\nHashMap after deleting all elements: " + hashMap);  
}  
}
```

**Q7. Design a program to store and display book details using ArrayList.**

**Book Details:**

**1. Book id**

**2. Book Name**

**3. Publication Year**

**4. Price**

```
import java.util.ArrayList;
import java.util.Scanner;

class Book {
    private int bookId;
    private String bookName;
    private int publicationYear;
    private double price;

    public Book(int bookId, String bookName, int publicationYear, double price) {
        this.bookId = bookId;
        this.bookName = bookName;
        this.publicationYear = publicationYear;
        this.price = price;
    }

    public int getBookId() {
        return bookId;
    }

    public String getBookName() {
        return bookName;
    }

    public int getPublicationYear() {
        return publicationYear;
    }

    public double getPrice() {
        return price;
    }
}
```

```

}

public void displayBookDetails() {

    System.out.println("Book ID: " + bookId);

    System.out.println("Book Name: " + bookName);

    System.out.println("Publication Year: " + publicationYear);

    System.out.println("Price: " + price);

    System.out.println("-----");

}

}

public class BookStore {

    public static void main(String[] args) {

        ArrayList<Book> bookList = new ArrayList<>();

        Scanner scanner = new Scanner(System.in);

        char choice;

        do {

            System.out.print("Enter Book ID: ");

            int bookId = scanner.nextInt();

            scanner.nextLine();

            System.out.print("Enter Book Name: ");

            String bookName = scanner.nextLine();

            System.out.print("Enter Publication Year: ");

            int publicationYear = scanner.nextInt();

            System.out.print("Enter Price: ");

            double price = scanner.nextDouble();

            Book book = new Book(bookId, bookName, publicationYear, price);

            bookList.add(book);

            System.out.print("Do you want to add another book? (y/n): ");

            choice = scanner.next().charAt(0);

        } while (choice == 'y' || choice == 'Y');

        System.out.println("\nList of Books:");
    }
}

```

```
for (Book book : bookList) {  
    book.displayBookDetails();  
}  
scanner.close();  
}  
}
```