1) ArrayList Operations

```
import java.util.*;

public class ArrayListOperations {
    public static void main(String[] args) {
        // create new ArrayList
        ArrayList<String> colors = new ArrayList<>();

        // add some colors
        colors.add("Red");
        colors.add("Blue");
        colors.add("Green");
        colors.add("Yellow");
        colors.add("Pink");

        System.out.println("Original ArrayList: " + colors);

        // iterate through all elements
        System.out.println("\nIterating elements:");
        for (String color : colors) {
            System.out.println(color);
        }

        // insert at first position
        colors.add(0, "Black");
        System.out.println("\nAfter inserting at first position: " + colors);

        // retrieve element at index 2
        System.out.println("\nElement at index 2: " + colors.get(2));

        // update specific element
        colors.set(1, "Orange");
        System.out.println("\nAfter updating 2nd element: " + colors);

        // remove third element
        colors.remove(2);
        System.out.println("\nAfter removing 3rd element: " + colors);

        // search element
        System.out.println("\nSearching 'Pink': " + colors.contains("Pink"));

        // sort ArrayList
        Collections.sort(colors);
        System.out.println("\nSorted ArrayList: " + colors);

        // copy into another list
        ArrayList<String> copiedList = new ArrayList<>(colors);
        System.out.println("\nCopied ArrayList: " + copiedList);

        // shuffle
        Collections.shuffle(colors);
        System.out.println("\nShuffled ArrayList: " + colors);

        // reverse
        Collections.reverse(colors);
        System.out.println("\nReversed ArrayList: " + colors);

        // extract portion (subList)
        List<String> subList = colors.subList(1, 4); // index 1 to 3
        System.out.println("\nExtracted portion: " + subList);

        // replace 2nd element
        colors.set(1, "White");
        System.out.println("\nAfter replacing 2nd element: " + colors);
    }
}
```

2) HashSet Set Operations

```
import java.util.*;

public class HashSetOperations {
    public static void main(String[] args) {
        HashSet<Integer> set1 = new HashSet<>(Arrays.asList(0, 1, 2, 3, 4, 5));
        HashSet<Integer> set2 = new HashSet<>(Arrays.asList(0, 1, 3, 4, 7, 8, 9));

        System.out.println("Set1: " + set1);
        System.out.println("Set2: " + set2);

        // Union
        HashSet<Integer> union = new HashSet<>(set1);
        union.addAll(set2);
        System.out.println("\nUnion: " + union);

        // Intersection
        HashSet<Integer> intersection = new HashSet<>(set1);
        intersection.retainAll(set2);
        System.out.println("Intersection: " + intersection);

        // Difference (set2 - set1)
        HashSet<Integer> difference = new HashSet<>(set2);
        difference.removeAll(set1);
        System.out.println("Difference: " + difference);
    }
}
```

3) HashMap with Employee Objects

```
import java.util.*;

class Employee {
    int id;
    String name;
    String city;

    public Employee(int id, String name, String city) {
        this.id = id;
        this.name = name;
        this.city = city;
    }
    public String toString() {
        return "Employee[ID=" + id + ", Name=" + name + ", City=" + city + "]";
    }
}

public class EmployeeHashMap {
    public static void main(String[] args) {
        Map<Integer, Employee> employeesMap = new HashMap<>();

        // Add employees
        employeesMap.put(101, new Employee(101, "Rohit", "Noida"));
        employeesMap.put(102, new Employee(102, "Aman", "Delhi"));
        employeesMap.put(103, new Employee(103, "Priya", "Mumbai"));
        employeesMap.put(104, new Employee(104, "Sneha", "Pune"));
        employeesMap.put(105, new Employee(105, "Vikas", "Bangalore"));

        // Display all employees
        System.out.println("All Employees:");
        for (Map.Entry<Integer, Employee> entry : employeesMap.entrySet()) {
            System.out.println("Key: " + entry.getKey() + " -> " + entry.getValue());
        }

        // Display employee by ID
        int searchId = 103;
        System.out.println("\nDetails of Employee with ID " + searchId + ": " + employeesMap.get(searchId));
    }
}
```