Day 12 – Collections Framework in Java

Objective:

To understand the concept and use of the Java Collections Framework (JCF) for storing, managing, and manipulating groups of objects efficiently.

Content:

Today, I learned about the **Java Collections Framework**, which provides a unified architecture for representing and managing data collections such as lists, sets, and maps.
Collections are more flexible than arrays because they can dynamically grow or shrink in size.

---

1. Overview of Collections Framework

The Java Collections Framework is part of the `java.util` package and includes **interfaces**, **classes**, and **algorithms** that handle data efficiently.

Important Interfaces:

- **List** – Ordered collection that allows duplicates.
- **Set** – Unordered collection that doesn't allow duplicates.
- **Map** – Stores data in key–value pairs.

---

2. Commonly Used Classes

Interface	Implementation Class	Description
List	`ArrayList`, `LinkedList`	Maintains ordered elements
Set	`HashSet`, `TreeSet`	Stores unique elements
Map	`HashMap`, `TreeMap`	Key-value storage

3. Example: Using ArrayList

```java
import java.util.*;
public class ListExample {
    public static void main(String[] args) {
        ArrayList<String> names = new ArrayList<>();
        names.add("Husanpreet");
        names.add("Gurpreet");
        names.add("Simran");
        for (String name : names) {
            System.out.println(name);
        }
    }
}
```

Output:

```
Husanpreet
Gurpreet
Simran
```

4. Example: Using HashMap

```java
import java.util.*;
public class MapExample {
    public static void main(String[] args) {
        HashMap<Integer, String> students = new HashMap<>();
        students.put(1, "Harsh");
        students.put(2, "Husan");
        students.put(3, "Simran");
        for (int id : students.keySet()) {
            System.out.println(id + " - " + students.get(id));
        }
    }
}
```

Output:

```
1 - Harsh
2 - Husan
3 - Simran
```

Learning Outcome:

Understood the need for collections to handle dynamic data efficiently.
Learned about List, Set, and Map interfaces and their common implementations.
Gained practical experience in using ArrayList and HashMap for storing and retrieving data.