

Task :4

# Java Collections

Date: 13-10-2025

Write a program to add Student details to a Hashtable. The key of the Hashtable will contain an object of the StudentID class, which stores the student id. The value in the Hashtable will be an object of StudentData class which contains details of an student like ID, name and CGPA

## Aim:

To write a Java program that stores student details in a Hashtable, where the key is an object of StudentID class containing the student ID, and the value is an object of StudentData class containing the student's details such as ID, name, and CGPA.

## Algorithm:

1. Package Name: *(optional)* mycollection – to organize collection classes.
2. Class Name:
  - o StudentID – stores student ID (used as key).
  - o StudentData – stores student details (ID, name, CGPA).
  - o StudentHashtable – contains the main() method to create a Hashtable and display data.
3. Method Name: main() – adds and retrieves student data from Hashtable.
4. Function:

Store key-value pairs in a Hashtable where:

  - o Key → StudentID object
  - o Value → StudentData object

## Program:

```
import java.util.*;  
  
/**  
 * Task 4: Java Collections - Hashtable Example  
 * Demonstrates adding and retrieving Student details using a Hashtable.  
 */  
  
// Class representing the Student ID (Key)  
class StudentID {  
    int id;  
  
    public StudentID(int id) {
```

```
        this.id = id;
    }

    // Override hashCode() and equals() for correct key comparison
    @Override
    public int hashCode() {
        return Objects.hash(id);
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        StudentID other = (StudentID) obj;
        return id == other.id;
    }
}

// Class representing Student Data (Value)
class StudentData {
    int id;
    String name;
    double cgpa;

    public StudentData(int id, String name, double cgpa) {
        this.id = id;
        this.name = name;
        this.cgpa = cgpa;
    }

    // Method to display student details
    public void display() {
        System.out.println("Student ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("CGPA: " + cgpa);
        System.out.println("-----");
    }
}

// Main class
public class StudentHashtable {
```

```
public static void main(String[] args) {
    // Create a Hashtable
    Hashtable<StudentID, StudentData> table = new Hashtable<>();

    // Add student details to Hashtable
    table.put(new StudentID(101), new StudentData(101, "Kishor G", 8.9));
    table.put(new StudentID(102), new StudentData(102, "Arun Kumar", 9.2));
    table.put(new StudentID(103), new StudentData(103, "Priya S", 8.5));

    // Display all student details
    System.out.println("----- Student Details from Hashtable -----");
    for (Map.Entry<StudentID, StudentData> entry : table.entrySet()) {
        entry.getValue().display();
    }
}
```

#### **Output:**

----- Student Details from Hashtable -----

Student ID: 101

Name: Kishor G

CGPA: 8.9

---

-----  
Student ID: 102

Name: Arun Kumar

CGPA: 9.2

---

-----  
Student ID: 103

Name: Priya S

CGPA: 8.5

---

#### **Result:**

Thus, a Java program using Hashtable was successfully implemented where the key (StudentID) stores student ID and the value (StudentData) stores details like ID, name, and CGPA.