CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Experiment - 6

Student Name: Vaibhav Kumar
UID: 23BCS13386
Branch: BE-CSE
Section/Group: KRG-2B
Semester: 5th
Date of Performance:14/10/25
Subject Name: Project Based Learning in Java
Subject Code: 23CSH-304

1. Aim:

Develop a Java program using lambda expressions and Stream operations to filter students scoring above 75%, sort them by marks, and display their names.

2. Objective:

To apply filtering, sorting, and transformation operations using the Stream API in Java for concise and efficient data processing.

3. Apparatus / Input Used:

- Programming Language: Java (JDK 8 or above)
- IDE: Eclipse / IntelliJ / VS Code
- Classes & Methods Used: Stream, filter(), sorted(), map(), collect()

4. Procedure:

1. Define a Student class with fields: name, id, and marks.
2. Create a list of student objects.
3. Use Stream API to:
   - Filter students with marks greater than 75.
   - Sort them by marks in descending order.
   - Extract and display their names.
4. Display the final list of students who scored above 75%.

5.Program Code:

```
import java.util.*;
import java.util.stream.*;
```

```
class Employee {
    String name;
    int id;
    double salary;

    Employee(String name, int id, double salary) {
        this.name = name;
        this.id = id;
        this.salary = salary;
    }

    public String toString() {
        return name + " - ₹" + salary;
    }
}

public class StreamEmployeeFilter {
    public static void main(String[] args) {

        List<Employee> employees = Arrays.asList(
            new Employee("Vaibhav", 201, 45000),
            new Employee("Riya", 202, 60000),
            new Employee("Arjun", 203, 52000),
            new Employee("Mehak", 204, 70000),
            new Employee("Karan", 205, 48000)
        );

        System.out.println("Employees earning above ₹50,000:");

        List<String> topEarners = employees.stream()
            .filter(e -> e.salary > 50000)
            .sorted((e1, e2) -> Double.compare(e2.salary, e1.salary))
            .map(e -> e.name)
            .collect(Collectors.toList());

        topEarners.forEach(System.out::println);
    }
}
```

6.Sample Output:

```
Output                                                    Clear
Employees earning above ₹50,000:
Mehak
Riya
Arjun
```