



## Experiment - 6

**Student Name:** Aditya Kumar Singh

**Branch:** BE-CSE

**Semester:** 5<sup>th</sup>

**Subject Name:** Project Based Learning in Java

**Subject Code:** 23CSH-304

**UID:** 23BCS11734

**Section/Group:** KRG-2B

**Date of Performance:** 14/10/25

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

### **Program Code:**

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

class Student {
    String name;
    int id;
    double marks;
```

```

Student(String name, int id, double marks) {
    this.name = name;
    this.id = id;
    this.marks = marks;
}

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

public class StreamStudentFilter {

    public static void main(String[] args) {
        List<Student> students = Arrays.asList(
            new Student("Ravi", 101, 85.5),
            new Student("Aditi", 102, 92.0),
            new Student("Kiran", 103, 78.0),
            new Student("Manoj", 104, 68.0),
            new Student("Tina", 105, 72.5)
        );
        System.out.println("Students scoring above 75%:");
        List<String> topStudents = students.stream()
            .filter(s -> s.marks > 75)
            .sorted((s1, s2) -> Double.compare(s2.marks, s1.marks))
            .map(s -> s.name)
            .collect(Collector.toList());
        topStudents.forEach(System.out::println);
    }
}

```

### **Sample Output:**

Students scoring above 75%:  
 Aditi  
 Ravi  
 Kiran