



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment – 6

Student Name: Hemant Narain Jha
Branch: BE-CSE
Semester: 5th
Subject Name: PBLJ

UID: 23BCS10022
Section/Group: KRG-2B
Date of Performance: 14/10/25
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: 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%.

Program Code:

```
import java.util.*;  
import java.util.stream.*;  
  
class Student {  
    String name; int id;  
    double marks;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
Student(String name, int id, double marks) {  
    this.name = name;  
    this.id = id; this.marks = marks;  
}  
public String toString() { return name + " - " + marks; }  
}  
  
public class EXPERIMENT6 {  
    public static void main(String[] args) {  
        List<Student> students = Arrays.asList(  
            new Student("Himanshu", 201, 89.3),  
            new Student("Neha", 202, 95.7),  
            new Student("Arjun", 203, 76.8),  
            new Student("Priya", 204, 82.1),  
            new Student("Vikram", 205, 67.5)  
        );  
  
        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(Collectors.toList());  
  
        topStudents.forEach(System.out::println);  
    }  
}
```

Sample Output:

```
● PS C:\Users\ASUS\Desktop\Sem 5\PBLJ_23BCS10889_KRG-2B> & 'C:\Program  
  '-cp' 'C:\Users\ASUS\AppData\Roaming\Code\User\workspaceStorage\b6d4a  
f85b6\bin' 'EXPERIMENT6'  
Students scoring above 75:  
Neha  
Himanshu  
Priya  
Arjun  
○ PS C:\Users\ASUS\Desktop\Sem 5\PBLJ_23BCS10889_KRG-2B>
```