Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session 2025-2026

Vision: Dream of where you want.	**Mission:** Means to achieve Vision

Program Educational Objectives of the program (PEO): (broad statements that describe the professional and career accomplishments)

PEO1	**Preparation**	**P: Preparation**	**Pep-CL abbreviation pronounce as Pep-si-lL easy to recall**
PEO2	**Core Competence**	**E: Environment (Learning Environment)**	
PEO3	**Breadth**	**P: Professionalism**	
PEO4	**Professionalism**	**C: Core Competence**	
PEO5	**Learning Environment**	**L: Breadth (Learning in diverse areas)**	

Program Outcomes (PO): (statements that describe what a student should be able to do and know by the end of a program)

Keywords of POs:
Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

PSO Keywords: Cutting edge technologies, Research

"I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life." *to contribute to the development of cutting-edge technologies and Research*.

Integrity: I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

Name and Signature of Student and Date
(Signature and Date in Handwritten)
Nikhil Gourkar

Gourkar

Nagar Yuwak Shikshan Sanstha's

Yeshwantrao Chavan College of Engineering

(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session	**2025-26 (ODD)**	**Course Name**	**Lab : Java Stack**
Semester	**5**	**Course Code**	**CT**
Roll No	**59**	**Name of Student**	**Nikhil Gourkar**

Practical Number	**3**
Course Outcome	Proper Understanding of Basic Java programs covering loops , arrays and conditionals and implementation of programs.
Aim	To implement OOPS methods like polymorphism , method overloading and overriding .
Problem Definition	**1.Problem Statement:** Build a mini shopping cart system. Requirements are 1. Use a HashMap<String, Integer>; to store items (key = item name, value = quantity). 2. Allow adding items to the cart. 3. If the user tries to add an item with negative quantity, throw a custom exception InvalidQuantityException. 4. Display the final cart contents. **2.Problem Statement:** A college system needs to store student grades. Requirements are 1. Use a TreeMap<;String, Integer>; where key = student name, value = marks. 2. Add multiple students with marks. 3. If someone tries to insert marks greater than 100 or less than 0, throw an exception InvalidMarksException. 4. Display students in sorted order of names (TreeMap will handle this). 5. Print the top scorer.
Theory (100 words)	**HashMap** A HashMap in Java is a collection that stores data in key–value pairs using hashing. It provides very fast performance with average constant time complexity O(1) for insertion, deletion, and lookup. However, it does not maintain any order of the keys. HashMap allows one null key and multiple null values. Internally, it uses a hash table and handles collisions using linked lists or balanced trees (from Java 8 onwards). It is best suited when fast access to data is required without caring about order. **TreeMap** A TreeMap, on the other hand, is a map that stores data in a sorted

<table>
<tr><td></td><td>order of keys. It is implemented using a Red-Black Tree (a self-balancing binary search tree). All operations like insertion, deletion, and lookup take O(log n) time. Unlike HashMap, TreeMap does not allow null keys, but it does allow multiple null values. TreeMap is useful when data needs to be maintained in a sorted sequence or when range-based queries are required.</td></tr>
<tr><td>Procedure and Execution

(100 Words)</td><td>Algorithm:
<u>Hashmap</u>
1. Initialize: Create an empty HashMap (key = item name, value = quantity).
2. Input Item: Get the item name and quantity from the user.
3. Check Quantity: If quantity < 0, throw an exception.
4. Add to HashMap:
5. If the item already exists key in the map, get the existing quantity.
6. Add the new quantity to the existing quantity.
7. Put the updated quantity back into the map.
8. Repeat steps 2-4 until input ends.
9. Display HashMap: Iterate over all items and print their values.
<u>TreeMap</u>
1. Initialize: Create an empty TreeMap (key = student name, value = marks).
2. Input Student and Marks: Get student name and marks from user.
3. Validate Marks: If marks < 0 or marks > 100, throw an exception.
4. Add to TreeMap:
5. Insert student name and marks. TreeMap automatically keeps keys sorted.
6. Repeat steps 2-4 until input ends.
7. Display TreeMap: Iterate over entries (in sorted order by key) and print.
8. Find Top Scorer: Traverse all entries to find student with max marks and print.</td></tr>
<tr><td></td><td>Code:

```
1) import java.util.*;

class InvalidQuantityException extends Exception {
   public InvalidQuantityException(String msg) {
      super(msg);
   }
}
public class Practical3A{
   public static void main(String[] args) {
      HashMap<String, Integer> cart = new HashMap<>();
      Scanner sc = new Scanner(System.in);
      while(true) {
         System.out.print("\nItem name (or end): ");
         String name = sc.nextLine();
         if(name.equals("end")) break;
```

</td></tr>
</table>

```
            System.out.print("Quantity: ");
            int qty = sc.nextInt();
            sc.nextLine();
            try {
                if(qty < 0) throw new InvalidQuantityException("negative
qty");
                cart.put(name, cart.getOrDefault(name, 0) + qty);
            } catch(Exception e) {
                System.out.println(e.getMessage());
            }
        }
        for(String k : cart.keySet()) {
            System.out.println(k + " " + cart.get(k));
        }
    }
}
2) import java.util.*;

class InvalidMarksException extends Exception {
    public InvalidMarksException(String msg) {
        super(msg);
    }
}

public class Practical3B{
    public static void main(String[] args) {
        TreeMap<String, Integer> students = new TreeMap<>();
        Scanner sc = new Scanner(System.in);
        while(true) {
            System.out.print("Student (or end): ");
            String name = sc.nextLine();
            if(name.equals("end")) break;
            System.out.print("Marks: ");
            int marks = sc.nextInt();
            sc.nextLine();
            try {
                if(marks < 0 || marks > 100) throw new
InvalidMarksException("Invalid marks");
                students.put(name, marks);
            } catch(Exception e) {
                System.out.println(e.getMessage());
            }
        }
        System.out.println("--- Sorted List ---");
        for(String k : students.keySet()) {
            System.out.println(k + " " + students.get(k));
        }
        String topName = ""; int topMarks = -1;
        for(String k : students.keySet()) {
            if(students.get(k) > topMarks) {
                topMarks = students.get(k); topName = k;
            }
        }
```

```
        if(topMarks != -1)
            System.out.println("Top Scorer: " + topName + " " +
topMarks);
    }
}
```

```
Item name (or end): Apple
Quantity: 3
Item name (or end): Banana
Quantity: 2
Item name (or end): Apple
Quantity: 1
Item name (or end): Orange
Quantity: -2
negative qty
Item name (or end): Bannana
Quantity: 2
Item name (or end): end
Apple 4
Bannana 2
Banana 2
```

```
Student (or end): Mithilesh
Marks: 67
Student (or end): Harshal
Marks: 10000
Invalid marks
Student (or end): Nkhil
Marks: 102
Invalid marks
Student (or end): Vedant
Marks: 100
Student (or end): end
--- Sorted List ---
Mithilesh 67
Vedant 100
Top Scorer: Vedant 100
```

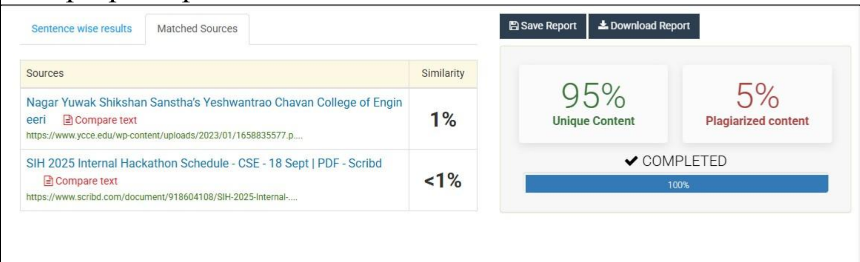
Output Analysis	1) When testing the mini shopping cart system, various scenarios happen during input and output. For example, when a user adds items like "Apple" with quantity 3, "Banana" with quantity 2, and adds "Apple" again with quantity 1, the system updates the cart so that "Apple" now shows quantity 4 and "Banana" 2. If the user tries to add "Orange" with a negative quantity, the custom exception triggers, showing an error without updating the cart. At the end, the list shows all items and their total quantities, such as "Apple 4", "Banana 2". This confirms the cart correctly accumulates quantities and rejects invalid input. 2) Similarly, for the college grades system, when students like "Alice" with 95 marks and "Bob" with 102 marks are entered, the system accepts Alice's marks but rejects Bob's marks as invalid. If "Charlie" with 88 is added, the TreeMap keeps the names sorted, showing "Alice 95" then "Charlie 88". Finally, the system finds and prints the top scorer, "Alice 95". This shows the program correctly validates input, keeps data sorted, and identifies the highest scorer.
Link of student Github profile where lab assignment has	https://github.com/Nikhil07Gourkar/Java-Lab

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

been uploaded	
Conclusion	The mini shopping cart and student grades systems handle input smoothly, adding valid entries and rejecting invalid ones with clear errors. The shopping cart totals quantities correctly, while the college system keeps students sorted and identifies the top scorer. Both programs demonstrate simple, effective use of Java collections with proper input validation.
Plag Report (Similarity index < 12%)	
Date	28/09/25