Experiment -5

Student Name: Abhishek Thakur **UID: 22BCS16176**

Branch: BE-CSE **Section/Group:22BCS_IOT-641/A**

Semester: 6th **Date of Performance: /02/2025**

Subject Name: Java **Subject Code: 22CSP-351**

PROBLEM – 1

1. **AIM:** Write a Java program to calculate the sum of a list of integers using autoboxing and unboxing. Include methods to parse strings into their respective wrapper classes (e.g., Integer.parseInt())

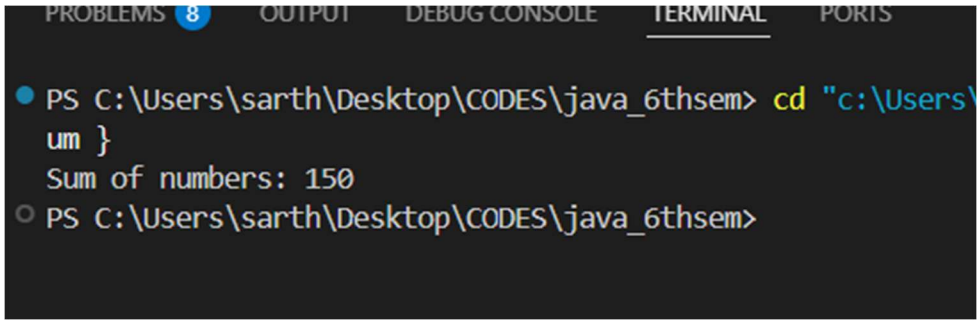
2. **Objective:**This program demonstrates autoboxing and unboxing by converting numeric strings into Integer objects and efficiently calculating their total sum.

3. Implementation:

```
import java.util.*;

public class AutoboxingSum {
   public static void main(String[] args) {
      List<Integer> numbers = new ArrayList<>();
      String[] strNumbers = {"10", "20", "30", "40", "50"};

      for (String str : strNumbers) {
         numbers.add(parseInteger(str)); // Autoboxing
      }

      int sum = calculateSum(numbers);

      System.out.println("Sum of numbers: " + sum);
   }

   public static Integer parseInteger(String str) {
      return Integer.parseInt(str); // Autoboxing
   }

   public static int calculateSum(List<Integer> numbers) {
      int sum = 0;
      for (Integer num : numbers) {
         sum += num; // Unboxing
      }
```

```
        return sum;
    }
}

}
```

4. OUTPUT:



PROBLEM2:

1. **AIM:** Create a Java program to serialize and deserialize a Student object. The program should: Serialize a Student object (containing id, name, and GPA) and save it to a file.Deserialize the object from the file and display the student details.Handle FileNotFoundException, IOException, and ClassNotFoundException using exception handling.

2. **Objective:**The objective of this Java program is to implement serialize a Student object with details such as id, name, and GPA and handle exceptions like FileNotFoundException, IOException, and ClassNotFoundException.

3. Implementation/Code:

```
import java.io.*;

class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    private int id;
    private String name;
    private double gpa;

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

    @Override
    public String toString() {
        return "Student{id=" + id + ", name='" + name + "', gpa=" + gpa + "}";
    }
}
```

```
public class StudentSerialization {
    private static final String FILE_NAME = "student.ser";

    public static void main(String[] args) {
        Student student = new Student(13692, " Abhishek  Th   ", 7.8);
        serializeStudent(student);
        Student deserializedStudent = deserializeStudent();
        if (deserializedStudent != null) {
            System.out.println("Deserialized Student: " + deserializedStudent);
        }
    }

    public static void serializeStudent(Student student) {
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
            oos.writeObject(student);
            System.out.println("Student serialized successfully.");
        } catch (IOException e) {
            System.err.println("Error during serialization: " + e.getMessage());
        }
    }

    public static Student deserializeStudent() {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            return (Student) ois.readObject();
        } catch (FileNotFoundException e) {
            System.err.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error during deserialization: " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.err.println("Class not found: " + e.getMessage());
        }
        return null;
    }
}
```

4. OUTPUT:

```
PROBLEMS 8    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\sarth\Desktop\CODES\java_6thsem> cd "c:\Users\sarth\Desktop\CODES\ja
entSerialization }
Student serialized successfully.
Deserialized Student: Student{id=13692, name='SARTHAK AG', gpa=7.8}
PS C:\Users\sarth\Desktop\CODES\java_6thsem>
```

5. Learning Outcomes:

a. **Problem-Solving Skills:** Understanding and applying problem-solving strategies for different scenarios, such as finding sums, minimizing jumps, simplifying paths, and implementing data structures using constraints.
b. **Algorithm Design:** Gained knowledge of:
    - Brute-force approach for finding pairs in an array.
    - Greedy algorithm for determining minimum jumps in an array.
    - Stack-based approach for resolving Unix-style paths.
    - Utilizing stacks to mimic the behavior of a queue.
c. **Data Structure Usage:**
    - Learned how to use arrays, stacks, and deques effectively to solve specific tasks.
    - Understood how two stacks can be combined to simulate a queue.
d. **Java Proficiency:**
    - Gained hands-on experience implementing solutions using Java, including class structures, loops, and stack operations.
e. **Complexity Analysis:**
    - Developed the ability to evaluate time and space complexity of algorithms.
    - Recognized the trade-offs between brute-force solutions and optimized approaches)