Experiment 5

Student Name: Sajal Saini **UID: 22BCS14505**
Branch: CSE **Section/Group: NTPP_601-B**
Semester: 6th **Date of Performance: 05/03/25**
Subject Name: Project based learning java **Subject Code:22CSH-359**

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()).

Source Code:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class AutoboxingUnboxingSum {

    public static List<Integer> parseStringToIntList(String[] strNumbers) {
        List<Integer> intList = new ArrayList<>();
        for (String num : strNumbers) {
            intList.add(Integer.parseInt(num));
        }
        return intList;
    }

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

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter numbers separated by spaces:");
        String input = scanner.nextLine();

        String[] strNumbers = input.split("\\s+");
        List<Integer> numbers = parseStringToIntList(strNumbers);
```

```
        int sum = calculateSum(numbers);
        System.out.println("Sum of the numbers: " + sum);
        scanner.close();
    }
```

}Screenshot of Outputs:

```
Enter numbers separated by spaces:
10 20 30 40 50 60
Sum of the numbers: 210

=== Code Execution Successful ===
```

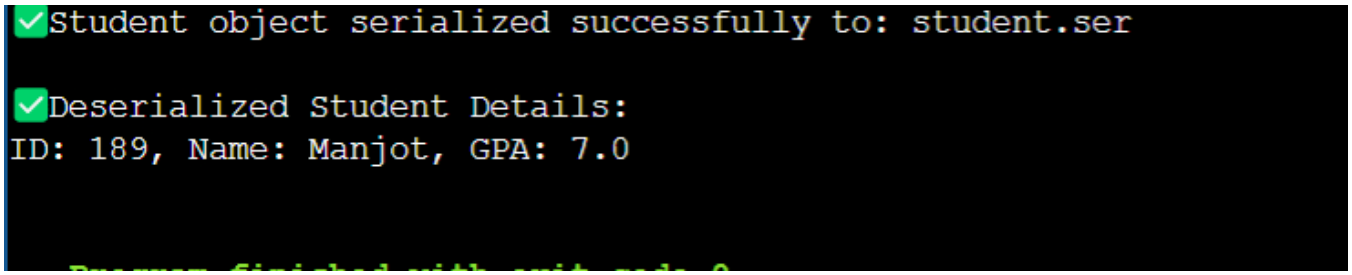
Aim: Create a Java program to serialize and deserialize a Student object. The program should: Serialize a Student object (containing id, name, and GPA) and save it to a file. Deserialize the object from the file and display the student details. Handle FileNotFoundException, IOException, and ClassNotFoundException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

Source Code:

```
import java.io.*;

class Student implements Serializable {
    private static final long serialVersionUID = 1L; // Ensures version compatibility
    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;
    }

    public void display() {
        System.out.println("ID: " + id + ", Name: " + name + ", GPA: " + gpa);
    }
}

public class Main {
    public static void main(String[] args) {
        Student student = new Student(189, "Manjot", 7.0);
        String filename = "student.ser"; // Change path if needed: "C:\\Users\\YourName\\student.ser"

        // Serialization
        try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(filename))) {
            out.writeObject(student);
```

```
            System.out.println(" Student object serialized successfully to: " + filename);
        } catch (IOException e) {
            System.err.println(" Serialization error: " + e.getMessage());
            e.printStackTrace();
        }

        // Deserialization
        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(filename))) {
            Student deserializedStudent = (Student) in.readObject();
            System.out.println("\n Deserialized Student Details:");
            deserializedStudent.display();
        } catch (FileNotFoundException e) {
            System.err.println(" File not found: " + filename);
        } catch (IOException e) {
            System.err.println(" IO Exception occurred:");
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            System.err.println(" Class not found: " + e.getMessage());
        }
    }
}
```

Source Code:



Experiment 4

Student Name: Sajal Saini **UID: 22BCS14505**

Branch: BE-CSE **Section/Group: NTPP_601-B**

Semester:6th **Date of Performance: 13/02/2025**

Subject Name: Project Based Learning in Java with Lab **Subject Code: 22CSH-359**

1. **Aim: Write a Program to perform the basic operations like insert, delete, display and search in list. List contains String object items where these operations are to be performed.**
2. **Objective**: The objective of this program is to implement basic operations (insert, delete, display, and search) on a List containing String objects. The program will demonstrate how to manipulate a list using common list operations in Java, providing functionality to manage and interact with data stored in the list.

3. **Implementation/Code:**

```
import java.util.ArrayList;
import java.util.Scanner;

public class StringListOperations { private
    static ArrayList<String> list = new ArrayList<>();
    public static void insertItem(String item) { list.add(item);
    }

    public static void deleteItem(String item) { if
            (list.contains(item)) {
        list.remove(item);
            System.out.println(item + " has been removed.");
        } else {
```

```
        System.out.println(item + " not found in the list."); } }

public static void displayList() { if
    (list.isEmpty()) {
        System.out.println("The list is empty.");
    } else {
        System.out.println("List items: " + list); }
}

public static void searchItem(String item) { if
    (list.contains(item)) {
        System.out.println(item + " is found in the list."); }
    else {
        System.out.println(item + " is not found in the list."); }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int choice; do {
        System.out.println("\nSelect an operation:");
        System.out.println("1. Insert Item");
        System.out.println("2. Delete Item");
        System.out.println("3. Display List");
        System.out.println("4. Search Item");
        System.out.println("5. Exit"); choice
        = sc.nextInt(); sc.nextLine();

        switch (choice) { case
            1:
                System.out.print("Enter item to insert: "); String
                    insertItem = sc.nextLine();
                insertItem(insertItem); break;
            case 2:
                System.out.print("Enter item to delete: "); String
                    deleteItem = sc.nextLine();
                deleteItem(deleteItem); break;
```

```
                case 3:
                   displayList();
                   break;
                case 4:
                   System.out.print("Enter item to search: ");
                String searchItem = sc.nextLine(); searchItem(searchItem);
                break; case 5:
                   System.out.println("Exiting program."); break;  default:
                   System.out.println("Invalid choice! Please choose a valid option."); }
            } while (choice != 5);
sc.close();
        }
        }
```

4. **Output:**

```
Select an operation:
1. Insert Item
2. Delete Item
3. Display List
4. Search Item
5. Exit
1
Enter item to insert: apple

Select an operation:
1. Insert Item
2. Delete Item
3. Display List
4. Search Item
5. Exit
3
List items: [apple]

Select an operation:
1. Insert Item
2. Delete Item
3. Display List
4. Search Item
5. Exit
```

5. **Learning Outcomes:**
    1. Learn how to perform basic **CRUD (Create, Read, Update, Delete)** operations on a **List** of **String** objects in Java.
    2. Understand how to use the **ArrayList** class for dynamically storing and manipulating a collection of items.

3. Practice handling **user input** using the **Scanner** class for interaction with the program.
4. Implement methods for **searching**, **deleting**, and **displaying** items in a list efficiently.
5. Gain familiarity with **control flow** and **loops** to allow for continuous user interaction until the program is exited.