

EXPERIMENT -5

1. 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()).(Easy)

Code –

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

public class AutoBoxingExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Integer> numbers = new ArrayList<>();

        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        System.out.println("Enter " + n + " numbers:");
        for (int i = 0; i < n; i++) {
            numbers.add(scanner.nextInt()); // Autoboxing from int to Integer
        }

        int sum = 0;
        for (Integer num : numbers) {
            sum += num; // Unboxing Integer to int
        }

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

        // Parsing Strings into Wrapper Classes
        scanner.nextLine(); // Consume newline
        System.out.print("Enter first number as string: ");
        String strNum1 = scanner.nextLine();

        System.out.print("Enter second number as string: ");
        String strNum2 = scanner.nextLine();

        int parsedSum = Integer.parseInt(strNum1) +
        Integer.parseInt(strNum2);
        System.out.println("Parsed Sum: " + parsedSum);

        scanner.close();
    }
}
```

OUTPUT-

✓ TERMINAL

```
● mrpkmehta@Priyamanshus-MacBook-Air easyLevel % javac AutoBoxingExample.java
● mrpkmehta@Priyamanshus-MacBook-Air easyLevel % java AutoBoxingExample
Enter the number of elements: 5
Enter 5 numbers:
4 6 7 8 4
Sum of numbers: 29
Enter first number as string: 5
Enter second number as string: 7
Parsed Sum: 12
○ mrpkmehta@Priyamanshus-MacBook-Air easyLevel % □
```

2. 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. (Medium)

Code –

```

import java.io.*;

// Student class must implement Serializable
class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    int id;
    String name;
    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 StudentSerialization {
    private static final String FILE_NAME = "student.ser";

    public static void main(String[] args) {
        Student student = new Student(17266, "Priyamanshu", 7.8);
        serializeStudent(student);
        deserializeStudent();
    }

    // Serialize Student object
    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.out.println("Serialization Error: " + e.getMessage());
        }
    }

    // Deserialize Student object
    public static void deserializeStudent() {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            Student student = (Student) ois.readObject();
            System.out.println("Deserialized Student Details:");
            student.display();
        } catch (FileNotFoundException e) {
            System.out.println("File not found!");
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Deserialization Error: " + e.getMessage());
        }
    }
}

```

OUTPUT-

▼ TERMINAL

```

● mrpkmehta@Priyamanshus-MacBook-Air mediumLevel % javac StudentSerialization.java
● mrpkmehta@Priyamanshus-MacBook-Air mediumLevel % java StudentSerialization
Student serialized successfully.
Deserialized Student Details:
ID: 17266, Name: Priyamanshu, GPA: 7.8
○ mrpkmehta@Priyamanshus-MacBook-Air mediumLevel % █

```

3. Create a menu-based Java application with the following options. 1.Add an Employee 2. Display All 3. Exit If option 1 is selected, the application should gather details of the employee like employee name, employee id, designation and salary and store it in a file. If option 2 is selected, the application should display all the employee details. If option 3 is selected the application should exit.(Hard)

Code –

```
import java.io.*;
import java.util.*;

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    int id;
    String name, designation;
    double salary;

    public Employee(int id, String name, String designation, double salary) {
        this.id = id;
        this.name = name;
        this.designation = designation;
        this.salary = salary;
    }

    public void display() {
        System.out.println("ID: " + id + ", Name: " + name + ", Designation: " + designation + ",
Salary: " + salary);
    }
}

public class EmployeeManagement {
    private static final String FILE_NAME = "employees.dat";
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\n1. Add Employee\n2. Display All\n3. Exit");
            System.out.print("Enter choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    addEmployee();
                    break;
                case 2:
                    displayEmployees();
                    break;
                case 3:
                    System.out.println("Exiting...");
                    return;
                default:
                    System.out.println("Invalid choice. Try again.");
            }
        }
    }
}
```

```

// Method to add an employee
private static void addEmployee() {
    System.out.print("Enter Employee ID: ");
    int id = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    System.out.print("Enter Name: ");
    String name = scanner.nextLine();

    System.out.print("Enter Designation: ");
    String designation = scanner.nextLine();

    System.out.print("Enter Salary: ");
    double salary = scanner.nextDouble();

    Employee emp = new Employee(id, name, designation, salary);

    try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME, true))) {
        oos.writeObject(emp);
        System.out.println("Employee added successfully!");
    } catch (IOException e) {
        System.out.println("Error saving employee: " + e.getMessage());
    }
}

// Method to display all employees
private static void displayEmployees() {
    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
        while (true) {
            Employee emp = (Employee) ois.readObject();
            emp.display();
        }
    } catch (EOFException e) {
        // End of file reached
    } catch (FileNotFoundException e) {
        System.out.println("No employee records found.");
    } catch (IOException | ClassNotFoundException e) {
        System.out.println("Error reading file: " + e.getMessage());
    }
}
}

```

OUTPUT-

TERMINAL

1. Add Employee

2. Display All

3. Exit

Enter choice: 1

Enter Employee ID: 17266

Enter Name: Priyamanshu

Enter Designation: HR

Enter Salary: 300000

Employee added successfully!

1. Add Employee

2. Display All

3. Exit

Enter choice: 2

ID: 234, Name: priyamanshu, Designation: HR, Salary: 200000.0

