EXPERIMENT-5

Name: Payal Kumari UID:22BCS10967 Section: 619-B

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

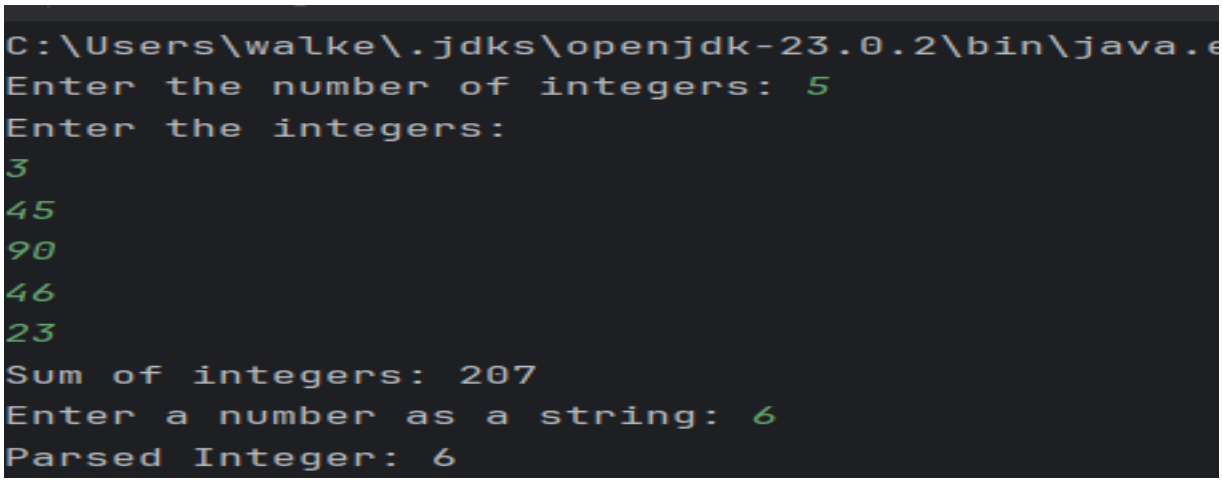
CODE:

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


public class AutoboxingUnboxing {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of integers: ");
        int n = scanner.nextInt();
        ArrayList<Integer> numbers = new ArrayList<>();
        System.out.println("Enter the integers: ");
        for (int i = 0; i < n; i++) {
            int num = scanner.nextInt();
            numbers.add(num);
        }
        int sum = 0;
        for (Integer num : numbers) {
            sum += num;
        }
        System.out.println("Sum of integers: " + sum);
        System.out.print("Enter a number as a string: ");
        String strNum = scanner.next();
        Integer parsedNum = Integer.parseInt(strNum);
        System.out.println("Parsed Integer: " + parsedNum);
```

```
        scanner.close();
    }
```

}**OUTPUT:**



Problem 2: Serialization and Deserialization of a Student Object in Java

This Java program demonstrates serialization and deserialization of a Student object

CODE:

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

class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    private String name;
    private int age;
    private String course;
    public Student(String name, int age, String course) {
        this.name = name;
        this.age = age;
        this.course = course;
    }
    public String toString() {
```

```
        return "Student{name='" + name + "', age=" + age + ", course='" + course + "'}";
    }
}

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

        System.out.print("Enter student name: ");
        String name = scanner.nextLine();
        System.out.print("Enter student age: ");
        int age = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter student course: ");
        String course = scanner.nextLine();

        Student student = new Student(name, age, course);

        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream("student.ser"))) {
            out.writeObject(student);
            System.out.println("Student object serialized successfully.");
        } catch (IOException e) {
            e.printStackTrace();
        }

        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream("student.ser"))) {
            Student deserializedStudent = (Student) in.readObject();
            System.out.println("Deserialized Student: " + deserializedStudent);
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
```

```
        scanner.close();
    }
}
```

OUTPUT:

```
C:\Users\walke\.jdks\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Progra
Enter student name: Payal
Enter student age: 21
Enter student course: CSE
Student object serialized successfully.
Deserialized Student: Student{name='Payal', age=21, course='CSE'}

Process finished with exit code 0
```

Problem 3: To create a menu-based Java application that allows the addition and display of employee details, and stores this information in a file for persistence.

CODE:

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

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    private int id;
    private String name;
    private String designation;
    private 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 String toString() {
        return "Employee ID: " + id + "\nName: " + name + "\nDesignation: " + designation +
"\nSalary: " + salary + "\n";
    }
}
```

```
public class EmployeeManagement {
    private static final String FILE_NAME = "employees.ser";

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

        while (true) {
            System.out.println("\nEmployee Management System");
            System.out.println("1. Add Employee");
            System.out.println("2. Display All Employees");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    addEmployee(scanner);
                    break;
                case 2:
                    displayEmployees();
                    break;
                case 3:
                    System.out.println("Exiting program...");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice! Please enter 1, 2, or 3.");
            }
        }
    }

    private static void addEmployee(Scanner scanner) {
        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 employee = new Employee(id, name, designation, salary);

        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(FILE_NAME, true))) {
            out.writeObject(employee);
            System.out.println("Employee added successfully!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static void displayEmployees() {
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            while (true) {
                Employee employee = (Employee) in.readObject();
                System.out.println(employee);
            }
        } catch (EOFException e) {
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

OUTPUT: