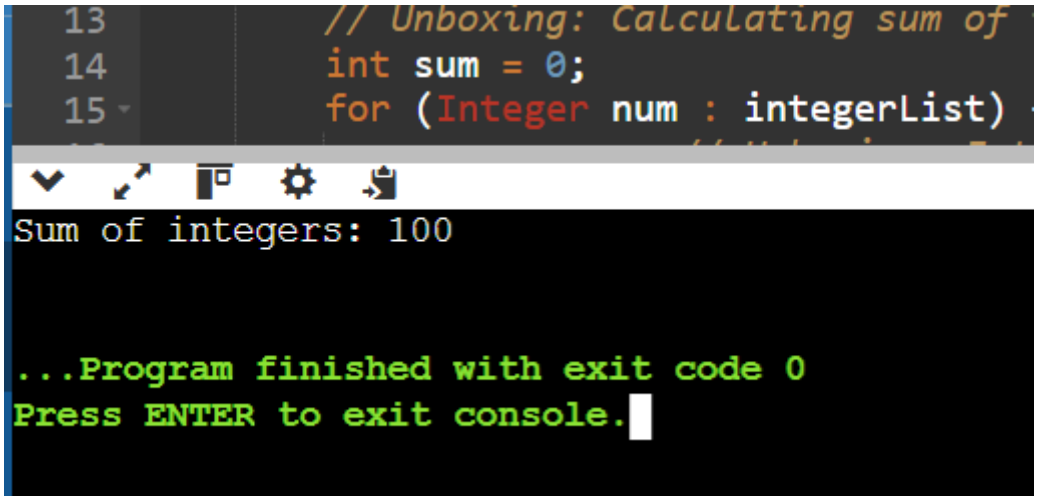
Experiment-5

Part-A: Autoboxing, Unboxing, and Sum Calculation

Code:

```
import java.util.*;
public class AutoboxingUnboxing {
    public static void main(String[] args) {
        String[] numberStrings = {"10", "20", "30", "40"};
        List<Integer> integerList = new ArrayList<>();
        for (String number : numberStrings) {
            integerList.add(Integer.parseInt(number));
        }
        int sum = 0;
        for (Integer num : integerList) {
            sum += num; // Unboxing: Integer to int
        }
        System.out.println("Sum of integers: " + sum);
    }
}
```

Output:



Part-B: Serialization and Deserialization of a Student Object

Code:

```
import java.io.*;
class Student implements Serializable {
   String name;
   int age;
   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) {
      Student student = new Student("John Doe", 22, "Computer Science");
      try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream("student.ser"))) {
         out.writeObject(student);
         System.out.println("Student 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();
        }
    }
}
```

Output:

```
Student serialized successfully!
Deserialized Student: Student{name='Nidhi', age=22, course='Computer Science'}


...Program finished with exit code 0
Press ENTER to exit console.
```

Part-C: Menu-Based Employee Management System

Code:

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


class Employee implements Serializable {
    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 String toString() {
        return "Employee ID: " + id + "\nName: " + name + "\nDesignation: " +
designation + "\nSalary: " + salary;
    }
}

public class EmployeeManagement {
    static final String FILE_NAME = "employees.ser";
    static Scanner sc = new Scanner(System.in);
    static List<Employee> employees = new ArrayList<>();

    public static void addEmployee() {
        System.out.print("Enter Employee ID: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Designation: ");
        String designation = sc.nextLine();
        System.out.print("Enter Salary: ");
        double salary = sc.nextDouble();

        employees.add(new Employee(id, name, designation, salary));
```

```
        saveToFile();
        System.out.println("Employee added successfully!");
    }

    public static void displayEmployees() {
        loadFromFile();
        if (employees.isEmpty()) {
            System.out.println("No employees found!");
        } else {
            for (Employee emp : employees) {
                System.out.println(emp + "\n");
            }
        }
    }

    public static void saveToFile() {
        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
            out.writeObject(employees);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void loadFromFile() {
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
```

```
            employees = (List<Employee>) in.readObject();
        } catch (IOException | ClassNotFoundException e) {
            employees = new ArrayList<>();
        }
    }

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

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

Output:

```
1. Add Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter Employee ID: 101
Enter Name: Nidhi
Enter Designation: SE
Enter Salary: 400000
Employee added successfully!

1. Add Employee
2. Display All Employees
3. Exit
Enter your choice: 2
Employee ID: 101
Name: Nidhi
Designation: SE
Salary: 400000.0
```