Experiment 5

Student Name: Akshit Dutt **UID: 22BCS16465**
Branch: B.E CSE **Section: IOT-643-A**
Semester: 6th **DOP:24/02/25**
Subject: PBLJ **Subject Code: 22CSH-359**

Aim:

Develop Java programs using autoboxing, serialization, file handling, and efficient data processing and management.

Problem Statement :

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()).

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.

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.

Program :

1. **Sum of a List of Integers Using Autoboxing & Unboxing:**

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

public class AutoboxingUnboxingSum {

    public static Integer parseStringToInteger(String str) {
        return Integer.parseInt(str);
    }

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

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

        for (String numStr : numberStrings) {
            numbers.add(parseStringToInteger(numStr));
        }

        int sum = calculateSum(numbers);
        System.out.println("The sum of the numbers is: " + sum);
    }
}
```

2. **Student Serialization & Deserialization:**

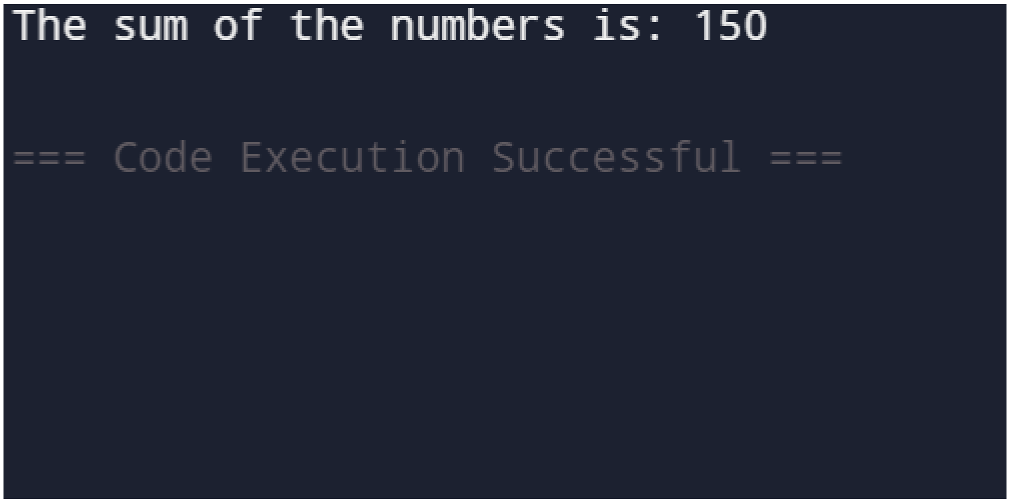
```
import java.io.*;

class Student implements Serializable {
    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 int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public double getGpa() {
        return gpa;
    }
}

public class StudentSerialization {

    public static void serializeStudent(Student student, String filename) {
        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(filename))) {
            out.writeObject(student);
            System.out.println("Student object serialized and saved to " + filename);
        } catch (FileNotFoundException e) {
            System.err.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("IOException occurred: " + e.getMessage());
        }
    }

    public static Student deserializeStudent(String filename) {
        Student student = null;
        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(filename))) {
            student = (Student) in.readObject();
            System.out.println("Student object deserialized from " + filename);
        } catch (FileNotFoundException e) {
            System.err.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("IOException occurred: " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.err.println("Class not found: " + e.getMessage());
        }
        return student;
    }
    public static void main(String[] args) {
        String filename = "student.ser";

        Student student1 = new Student(101, "Akshit Dutt", 6.85);

        serializeStudent(student1, filename);

        Student deserializedStudent = deserializeStudent(filename);

        if (deserializedStudent != null) {
            System.out.println("Student ID: " + deserializedStudent.getId());
            System.out.println("Student Name: " + deserializedStudent.getName());
            System.out.println("Student GPA: " + deserializedStudent.getGpa());
        }
    }
}
```

3. Employee Management System (Menu-Based) :

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

class Employee {
    private String name;
    private int id;
    private String designation;
    private double salary;

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

    public String getName() {
        return name;
    }

    public int getId() {
        return id;
    }

    public String getDesignation() {
        return designation;
    }
    public double getSalary() {
        return salary;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Designation: " + designation + ", Salary: " + 
salary;
    }
}

public class EmployeeManagement {

    private static final String FILE_NAME = "employees.dat";

    public static void addEmployee() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter employee name: ");
        String name = scanner.nextLine();
        System.out.print("Enter employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();  // consume the leftover newline
        System.out.print("Enter employee designation: ");
        String designation = scanner.nextLine();
        System.out.print("Enter employee salary: ");
        double salary = scanner.nextDouble();

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

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

    public static void displayEmployees() {
        try (ObjectInputStream in = new ObjectInputStream(new 
FileInputStream(FILE_NAME))) {
            System.out.println("\nEmployee Details:");
            while (true) {
                Employee employee = (Employee) in.readObject();
                System.out.println(employee);
            }
        } catch (EOFException e) {
            // End of file reached, no more employees to display
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Error reading employee data: " + e.getMessage());
        }
    }

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

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

            switch (choice) {
                case 1:
                    addEmployee();
                    break;
                case 2:
                    displayEmployees();
                    break;
                case 3:
                    System.out.println("Exiting the application...");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice, please try again.");
            }
        }
    }
}
```

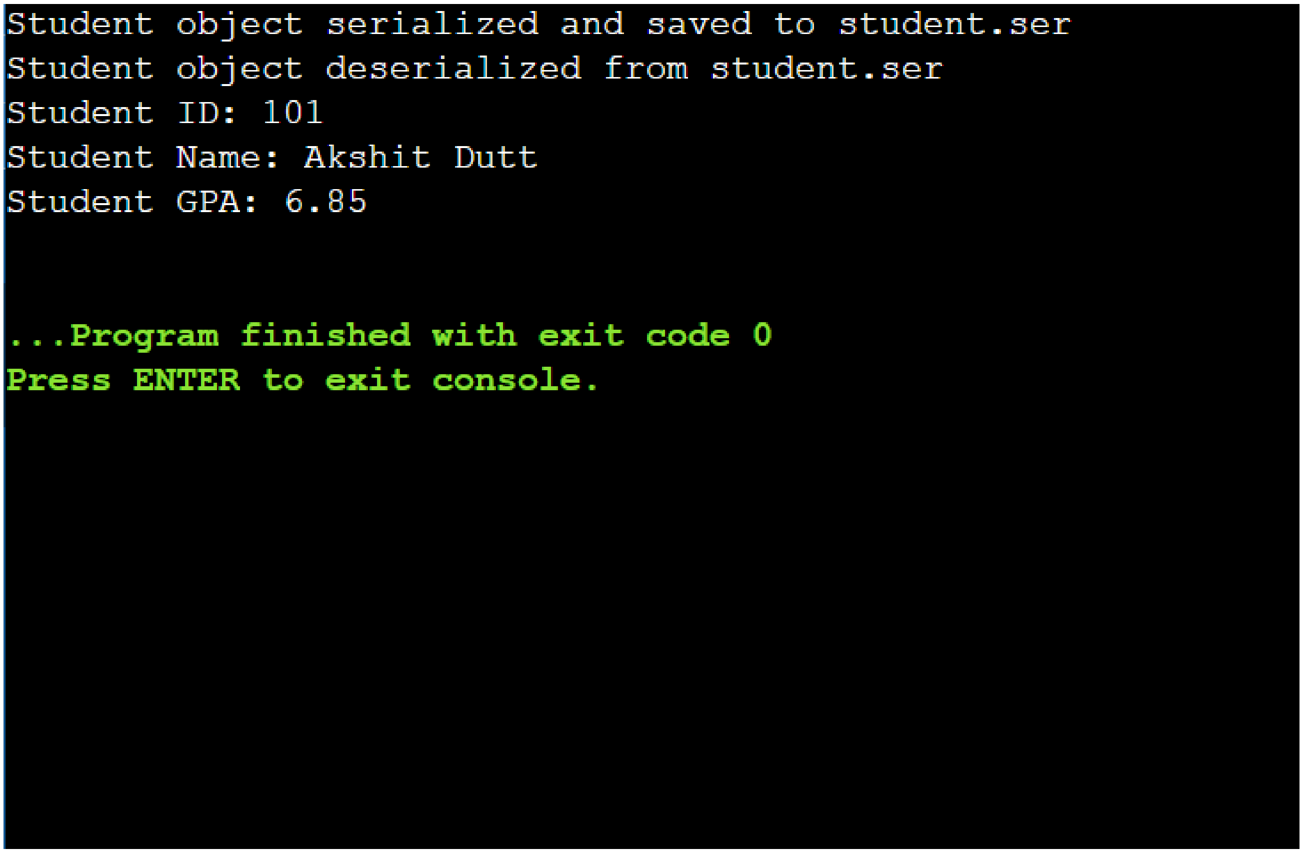
OUTPUT :

1. **Sum of a List of Integers Using Autoboxing & Unboxing:**



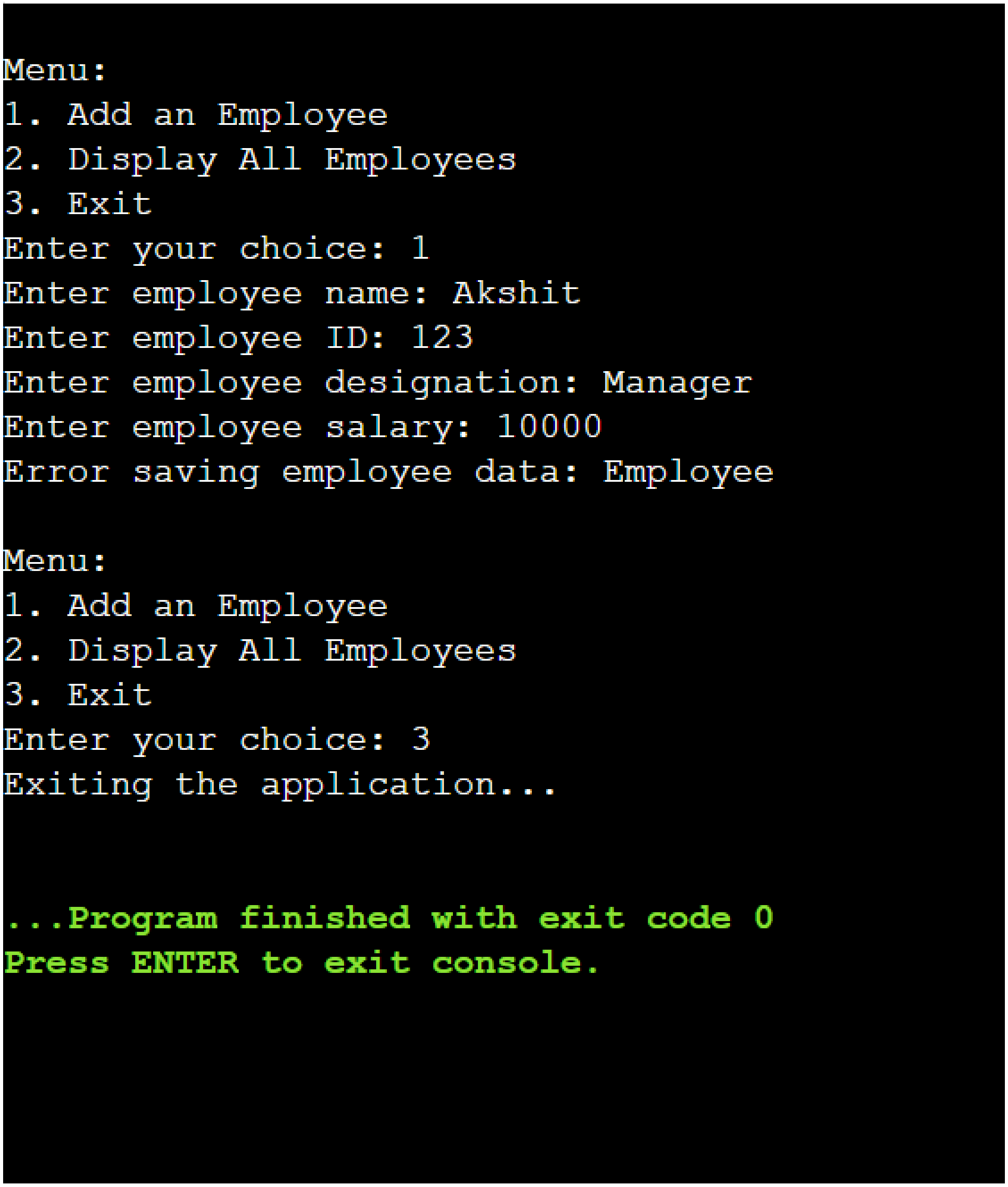
```
The sum of the numbers is: 150


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

2. **Student Serialization & Deserialization:**



```
Student object serialized and saved to student.ser
Student object deserialized from student.ser
Student ID: 101
Student Name: Akshit Dutt
Student GPA: 6.85


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

3. **Employee Management System (Menu-Based) :**



```
Menu:
1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter employee name: Akshit
Enter employee ID: 123
Enter employee designation: Manager
Enter employee salary: 10000
Error saving employee data: Employee

Menu:
1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 3
Exiting the application...


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

Learning Outcomes:

- **Implement object-oriented programming with classes, encapsulation, and serialization.**
- **Utilize core Java concepts like loops, conditionals, autoboxing, and unboxing.**
- **Apply file handling with serialization, deserialization, and exception management.**