Name :- Harsh Yadav

Sec 636/A

Q1.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 handlingQ1.

->

```
import java.io.*;
// Serializable Student class
class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    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("Student ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("GPA: " + gpa);
    }
}
public class StudentSerializationDemo {
    private static final String FILE_NAME = "student.ser";
    public static void serializeStudent(Student student) {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME))) {
            oos.writeObject(student);
            System.out.println("Student object serialized successfully.");
        } catch (FileNotFoundException e) {
```

Name :- Harsh Yadav

Sec 636/A

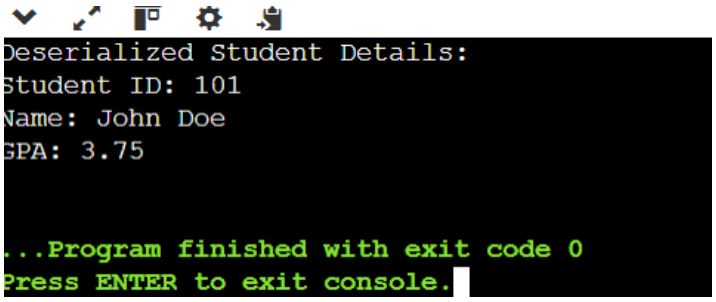
```
            System.err.println("Error: File not found. " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error: Unable to serialize object. " + e.getMessage());
        }
    }
    public static Student deserializeStudent() {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            return (Student) ois.readObject();
        } catch (FileNotFoundException e) {
            System.err.println("Error: File not found. " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error: Unable to deserialize object. " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.err.println("Error: Class not found. " + e.getMessage());
        }
        return null;
    }
    public static void main(String[] args) {
        Student student = new Student(101, "John Doe", 3.75);
        serializeStudent(student);

        Student deserializedStudent = deserializeStudent();
        if (deserializedStudent != null) {
            System.out.println("Deserialized Student Details:");
            deserializedStudent.display();
        }
    }
}
```

Name :- Harsh Yadav

Sec 636/A

OUTPUT



Q2. 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.

-> import java.io.*;

import java.util.*;

```
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 void display() {
        System.out.println("Employee ID: " + id);
```

Name :- Harsh Yadav

Sec 636/A

```
        System.out.println("Name: " + name);
        System.out.println("Designation: " + designation);
        System.out.println("Salary: " + salary);
        System.out.println("-------------------------");
    }
}

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

    public static void addEmployee(Scanner scanner) {
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Employee 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);
        employees.add(emp);
        saveEmployees();
        System.out.println("Employee added successfully!\n");
    }

    public static void displayEmployees() {
        loadEmployees();
```

Name :- Harsh Yadav

Sec 636/A

```
        if (employees.isEmpty()) {
            System.out.println("No employees found!\n");
        } else {
            System.out.println("Employee Details:");
            for (Employee emp : employees) {
                emp.display();
            }
        }
    }
    private static void saveEmployees() {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME))) {
            oos.writeObject(employees);
        } catch (IOException e) {
            System.err.println("Error saving employees: " + e.getMessage());
        }
    }
    private static void loadEmployees() {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            employees = (List<Employee>) ois.readObject();
        } catch (FileNotFoundException e) {
            employees = new ArrayList<>();
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Error loading employees: " + e.getMessage());
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        loadEmployees();
        while (true) {
            System.out.println("1. Add Employee");
```

Name :- Harsh Yadav

Sec 636/A

```
        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 the application...");
                scanner.close();
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice! Please enter a valid option.\n");
        }
    }
  }
}
```

OUTPUT

Name :- Harsh Yadav

Sec 636/A