**Practice-5.2**

Code:

import java.io.\*;

import java.nio.file.FileSystems;

import java.nio.file.Path;

import java.nio.file.Paths;

// Employee class with custom serialization

class Employee {

private String name;

private int id;

// Constructor

public Employee(String name, int id) {

this.name = name;

this.id = id;

}

// Getters

public String getName() {

return name;

}

public int getId() {

return id;

}

// Custom serialization

public void writeObject(ObjectOutputStream oos) throws IOException {

oos.writeUTF(name);

oos.writeInt(id);

}

// Custom deserialization

public void readObject(ObjectInputStream ois) throws IOException {

name = ois.readUTF();

id = ois.readInt();

}

// Override toString() for easy printing

@Override

public String toString() {

return "Employee{name='" + name + "', id=" + id + "}";

}

}

// CombinedTest class to demonstrate Path resolution and custom serialization

public class CombinedTest {

// Method to test Path and FileSystem

public static void testPath() {

// Create an instance of FileSystem

java.nio.file.FileSystem fs = FileSystems.getDefault();

// Resolve a Path instance from a directory and filename path

Path directory = Paths.get("C:/JavaProgramming");

Path filename = Paths.get("employees.txt");

Path fullPath = directory.resolve(filename);

// Print the constructed Path

System.out.println("Constructed Path: " + fullPath);

}

// Serialize the Employee object to file

public static void serializeData(Employee employee) {

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("employee.ser"))) {

employee.writeObject(oos);

System.out.println("Employee serialized successfully.");

} catch (IOException e) {

System.err.println("Serialization error: " + e.getMessage());

}

}

// Deserialize the Employee object from file

public static Employee deserializeData() throws ClassNotFoundException {

Employee employee = null;

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"))) {

employee = new Employee("", 0); // Create an empty employee

employee.readObject(ois);

} catch (IOException e) {

System.err.println("Deserialization error: " + e.getMessage());

}

return employee;

}

public static void main(String[] args) {

// Test Path and FileSystem

System.out.println("Testing Path and FileSystem:");

testPath();

// Create an Employee object

Employee emp = new Employee("John Doe", 12345);

// Test serialization

System.out.println("\nTesting Serialization:");

serializeData(emp);

// Test deserialization

System.out.println("\nTesting Deserialization:");

try {

Employee deserializedEmp = deserializeData();

if (deserializedEmp != null) {

System.out.println("Deserialized Employee: " + deserializedEmp);

}

} catch (ClassNotFoundException e) {

System.err.println("ClassNotFoundException: " + e.getMessage());

}

// Main limitations of the java.io package (commented for reference)

System.out.println("\nLimitations of java.io package:");

System.out.println("1. Performance issues compared to java.nio.file.");

System.out.println("2. Limited file operations (no support for file attributes, symbolic links).");

System.out.println("3. Error handling with checked exceptions (IOException).");

System.out.println("4. Lack of direct support for advanced I/O operations like non-blocking I/O.");

System.out.println("5. No direct support for advanced file system features.");

}

}

Output:

