EXPERIMENT 5

StudentName: Aman Chaudhary	**UID:**22BET10222
Branch:BE-IT	**Section/Group:**BET-701-A
Semester: 6	**DateofPerformance:**25.02.2025
Subject Name: Project Based Learning in Java	**Subject Code:** 22ITH-359

Aim:

WriteaJavaprogramtocalculatethesumofalistofintegersusingautoboxingandunboxing.Include methods to parse strings into their respective wrapper classes (e.g., Integer.parseInt()).

Objective:

TheobjectiveofthisJavaprogramistodemonstrateobjectserializationanddeserializationbycreating a Student class with fields for id, name, and GPA. The program will serialize an instance of Student and store it in a file, then deserialize the object back into memory and display its details. It will also include proper exception handling for FileNotFoundException, IOException, and ClassNotFoundException to ensure robustness in file operations.

Sourcecode:

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

publicclassAutoboxingUnboxingExample{

    publicstaticList<Integer>parseStringList(String[]stringNumbers){
        List<Integer> integerList = new ArrayList<>();
        for (String num : stringNumbers) {
            integerList.add(Integer.parseInt(num));
        }
        return integerList;
    }

    publicstaticintcalculateSum(List<Integer>numbers){ int
        sum = 0;
        for(Integernum:numbers){
            sum += num;
```

```
        }
        return sum;
    }

    publicstaticvoidmain(String[]args){
        String[] stringNumbers = {"10", "20", "30", "40", "50"};
        List<Integer>numbers=parseStringList(stringNumbers);
        int sum = calculateSum(numbers);
        System.out.println("Sum of numbers: " + sum);
    }
}
```

```
PS C:\Users\HP>  & 'C:\Users\HP\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.9.9-hotspot
'C:\Users\HP\AppData\Local\Temp\vscodesws_f183a\jdt_ws\jdt.ls-java-project\bin' 'AutoboxingU
Sum of numbers: 150
PS C:\Users\HP>
```

Aim:

Create a Java program to serialize and deserialize a Student object. The program should: Serialize a Studentobject(containingid,name,andGPA)andsaveittoafile.Deserializetheobjectfromthefile and display the student details. Handle FileNotFoundException, IOException, and ClassNotFoundException using exception handling. write a program and program name to save without comments write objective in paragrpah form.

Objective:

The objective of this program is to implement serialization and deserialization of a Student object in Java. The program will create a Student class containing attributes suchasid,name,andGPA.Itwillserializeaninstanceofthisclassandsaveittoafile, ensuringthattheobjectstateispreservedforfutureretrieval.Additionally,theprogram will deserialize the Student object from the file and display its details on the console. Properexceptionhandlingmechanismswillbeincorporatedtomanagepotentialerrors suchasFileNotFoundException,IOException,andClassNotFoundException,ensuring the program executes smoothly and robustly.

aa

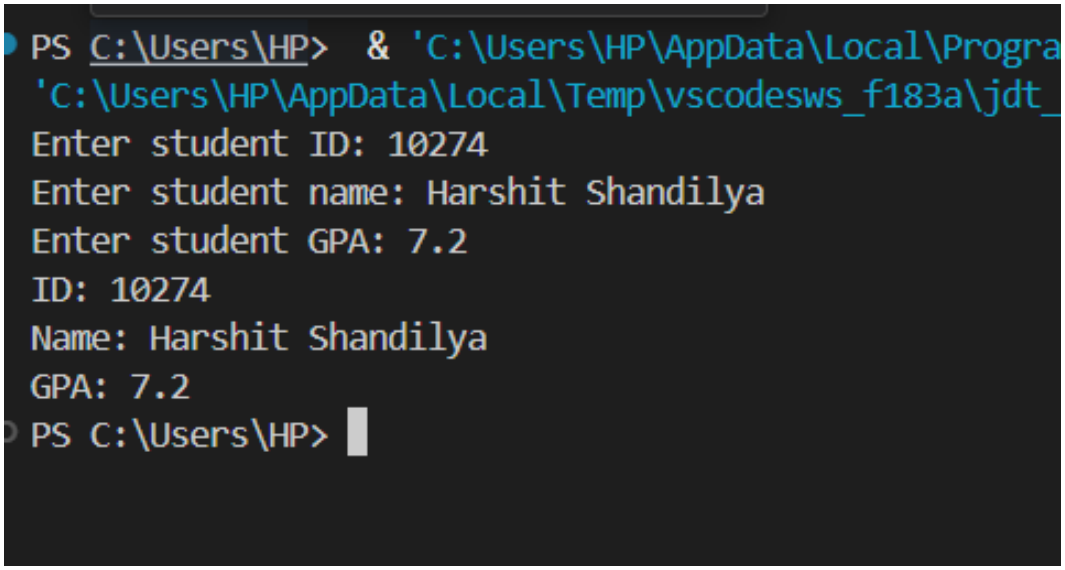
Sourcecode:

```
importjava.io.*;

importjava.util.Scanner;


classStudentimplementsSerializable{

   privatestaticfinallongserialVersionUID=1L;

   private int id;

   privateStringname; private

   double gpa;


   publicStudent(intid,Stringname,doublegpa){

      this.id = id;

      this.name =name;
```

```
        this.gpa= gpa;
    }


    public void display() {
        System.out.println("ID: " + id);
        System.out.println("Name:"+name);
        System.out.println("GPA: " + gpa);
    }
}


publicclassStudentSerialization{
    publicstaticvoidserializeStudent(Studentstudent,Stringfilename){ try
        (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(filename))){
            out.writeObject(student);
        }catch(IOExceptione){
            System.err.println("Errorduringserialization:"+e.getMessage());
        }
    }


    public static Student deserializeStudent(String filename) {
        try(ObjectInputStreamin=newObjectInputStream(new
FileInputStream(filename))){
            return(Student)in.readObject();
```

```
    } catch (FileNotFoundException e) {
        System.err.println("Filenotfound:"+e.getMessage());
    }catch(IOExceptione){
        System.err.println("Errorduringdeserialization:"+e.getMessage());
    } catch (ClassNotFoundException e) {
        System.err.println("Classnotfound:"+e.getMessage());
    }
    return null;
}

public static void main(String[] args) {
    Scannerscanner=newScanner(System.in);
    System.out.print("Enter student ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enterstudentname:");
    String name = scanner.nextLine();
    System.out.print("Enter student GPA: ");
    double gpa = scanner.nextDouble();

    Stringfilename="student.ser";
    Studentstudent=newStudent(id,name,gpa); serializeStudent(student,
    filename);
```

```
        StudentdeserializedStudent=deserializeStudent(filename); if
        (deserializedStudent != null) {
            deserializedStudent.display();
        }
    }
}
```



Aim:
Createamenu-basedJavaapplicationwiththefollowingoptions.1.AddanEmployee2. DisplayAll3.ExitIfoption1isselected,theapplicationshouldgatherdetailsofthe employeelikeemployeename,employeeid,designationandsalaryandstoreitinafile.If option2isselected,theapplicationshoulddisplayalltheemployeedetails.Ifoption3is selectedtheapplicationshouldexitwriteaprogramandprogramnametosavewithout comments.

Objective:
The objective of this program is to develop a menu-based Java application for managing employee records using file handling. The application provides three options: adding an employee, displaying all employees, and exiting. When adding an employee, the program collectsdetailssuchasemployeeID,name,designation,andsalary,andstorestheminafilefor persistent storage. When displaying employees, the program reads the stored employee records fromthefileandpresentsthemontheconsole.Theprogramensuresefficientdatahandlingand retrieval while incorporating proper exception handling to manage potential file-related error.

Sourcecode:

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

classEmployeeimplementsSerializable{
    privatestaticfinallongserialVersionUID=1L;
    private int id;
    private String name;
    privateStringdesignation;
    private double salary;

    publicEmployee(intid,Stringname,Stringdesignation,doublesalary){
        this.id = id;
        this.name = name;
        this.designation=designation;
        this.salary = salary;
    }

    public void display() {
        System.out.println("ID: " + id);
        System.out.println("Name:"+name);
        System.out.println("Designation:"+designation);
```

```
        System.out.println("Salary:"+salary);
    }
}

publicclassEmployeeManagement{
    privatestaticfinalStringFILE_NAME="employees.ser";

    publicstaticvoidaddEmployee(){
        try (Scanner scanner = new Scanner(System.in);
             ObjectOutputStreamout=newObjectOutputStream(new
FileOutputStream(FILE_NAME, true))) {
            System.out.print("EnterEmployeeID:");
            int id = scanner.nextInt();
            scanner.nextLine();
            System.out.print("EnterEmployeeName:");
            String name = scanner.nextLine();
            System.out.print("Enter Designation: ");
            String designation = scanner.nextLine();
            System.out.print("Enter Salary: ");
            doublesalary=scanner.nextDouble();
            Employeeemployee=newEmployee(id,name,designation,salary);
            out.writeObject(employee);
        }catch(IOExceptione){
            System.err.println("Errorwritingtofile:"+e.getMessage());
        }
    }

    publicstaticvoiddisplayEmployees(){
        try(ObjectInputStreamin=newObjectInputStream(new
FileInputStream(FILE_NAME))) {
            while(true){
                Employeeemployee=(Employee)in.readObject();
                employee.display();
                System.out.println();
            }
        }catch(EOFExceptione){
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Errorreadingfromfile:"+e.getMessage());
        }
    }
```

```
 public static void main(String[] args) {
    Scannerscanner=newScanner(System.in);
    while (true) {
       System.out.println("1.AddanEmployee");
       System.out.println("2. Display All");
       System.out.println("3. Exit");
       System.out.print("Choose an option: ");
       intchoice=scanner.nextInt();
       switch (choice) {
          case1:
             addEmployee();
             break;
          case2:
             displayEmployees();
             break;
          case3:
             System.exit(0);
          default:
             System.out.println("Invalidchoice.Pleasetryagain.");
       }
    }
 }
}
```

```
1. Add an Employee
2. Display All
3. Exit
Choose an option: 1
Enter Employee ID: 10274
Enter Employee Name: Harshit Shandilya
Enter Designation: Programmer
Enter Salary: 100000
```