Experiment5

StudentName:Harsh **UID:22BCS12301**
Branch:BE-CSE **Section/Group:22BCS639-A**
Semester: 6th **DateofPerformance:21/02/2025**
Subject Name: Project Based Learning inJavawithLab **Subject Code: 22CSH-359**

1. Aim: Develop Java programs using autoboxing, serialization, file handling, and efficientdata processing and management.

2. Objective: Todemonstrateautoboxing,unboxing,andcollectionhandlinginJava,along withobjectserializationanddeserializationwhileimplementingproperexceptionhandling. Additionally, to implement a menu-based employee management system using collections.

3. Implementation/Code:

WriteaJavaprogramtocalculatethesumofalistofintegersusingautoboxing andunboxing.Includemethodstoparsestringsintotheirrespectivewrapper classes (e.g., Integer.parseInt()).

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

public class SumUsingAutoboxing {
   publicstaticvoidmain(String[]args){
      List<Integer>numbers=newArrayList<>();
      numbers.add(parseInteger("10"));
      numbers.add(parseInteger("20"));
      numbers.add(parseInteger("30"));
      numbers.add(parseInteger("40"));
      numbers.add(parseInteger("50"));
      int sum = calculateSum(numbers);
      System.out.println("Sumofnumbers:"+sum);
}
   privatestaticIntegerparseInteger(Stringstr){ return
      Integer.parseInt(str);
 }
   privatestaticintcalculateSum(List<Integer>numbers){ int
      sum = 0;
      for(Integernum:numbers){ sum
         += num;
}
```

```
        returnsum;
    }
}
```

CreateaJavaprogramtoserializeanddeserializeaStudentobject.The program should:
SerializeaStudentobject(containingid,name,andGPA)andsaveittoafile. Deserializethe objectfromthe fileand displaythe student details.
HandleFileNotFoundException,IOException,andClassNotFoundExceptionusing exception handling.

```
importjava.io.*;

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("StudentID:"+id);
        System.out.println("Name: " + name);
        System.out.println("GPA: " + gpa);
}
}
publicclassStudentSerialization{
    privatestaticfinalStringFILE_NAME="student.ser"; public
    static void main(String[] args) {
        Studentstudent=newStudent(101,"ABCD",8.3);
        serializeStudent(student);
        deserializeStudent();
}
    privatestaticvoidserializeStudent(Studentstudent){
        try(ObjectOutputStreamoos=newObjectOutputStream(new
FileOutputStream(FILE_NAME))){
            oos.writeObject(student);
            System.out.println("Studentobjectserializedsuccessfully.");
        }
catch(IOExceptione){
            System.err.println("Errorduringserialization:"+e.getMessage());
```

```
 }
}
   privatestaticvoiddeserializeStudent(){
      try(ObjectInputStreamois=newObjectInputStream(new
FileInputStream(FILE_NAME))){
         Student student = (Student) ois.readObject();
         System.out.println("DeserializedStudentObject:");
         student.display();
      }
catch(FileNotFoundExceptione){
         System.err.println("Filenotfound:"+e.getMessage());
      }
catch(IOExceptione){
         System.err.println("Errorduringdeserialization:"+e.getMessage());
      }
catch(ClassNotFoundExceptione){
         System.err.println("Classnotfound:"+e.getMessage());
      }
   }
}
```

Createa menu-basedJavaapplicationwiththefollowingoptions.1.Add an Employee2.DisplayAll3.ExitIfoption1isselected,theapplicationshouldgather detailsoftheemployeelike employeename,employeeid,designationandsalary andstoreitinafile.Ifoption2isselected,theapplicationshoulddisplayallthe employeedetails.If option3 isselected the applicationshouldexit.

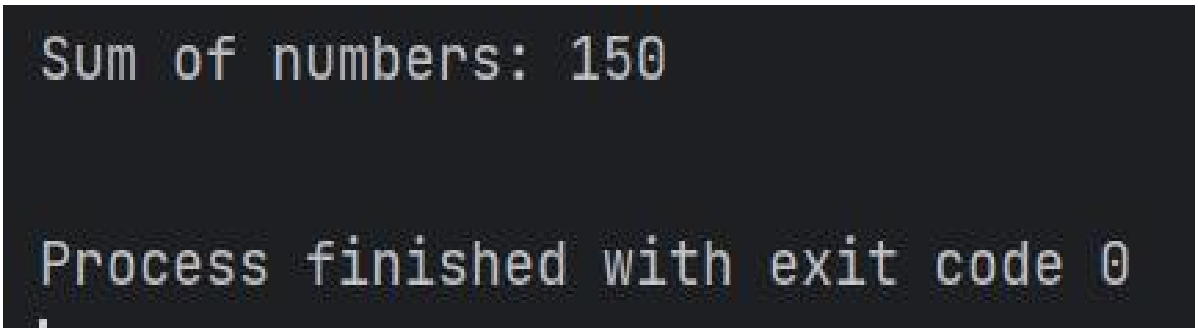
```
importjava.util.ArrayList;
import java.util.Scanner;
class Employee {
   intid;
   String name;
   Stringdesignation;
   double salary;
   publicEmployee(intid,Stringname,Stringdesignation,doublesalary){
      this.id = id;
      this.name = name;
      this.designation=designation;
      this.salary = salary;
   }
   @Override
   publicStringtoString(){
      return"ID:"+id+",Name:"+name+",Designation:"+designation+",Salary:"+ salary;
```

```
    }
}
public class EmployeeManagement {
    publicstaticvoidmain(String[]args){
        Scanner scanner = new Scanner(System.in);
        ArrayList<Employee>employees=newArrayList<>(); while
        (true) {
            System.out.println("\n1. Add an Employee");
            System.out.println("2.DisplayAllEmployees");
            System.out.println("3. Exit");

            System.out.print("Enteryourchoice:"); int
            choice = scanner.nextInt();
            scanner.nextLine();

            switch(choice){ case

                1:
                    System.out.print("EnterEmployeeID:"); int
                    id = scanner.nextInt(); scanner.nextLine();
                    // Consume newline
                    System.out.print("Enter Name: ");
                    String name = scanner.nextLine();
                    System.out.print("EnterDesignation:");
                    Stringdesignation=scanner.nextLine();
                    System.out.print("Enter Salary: ");
                    double salary = scanner.nextDouble();
                    employees.add(newEmployee(id,name,designation,salary));
                    System.out.println("Employee added successfully.");
                    break;

                case2:
                    if (employees.isEmpty()) {
                        System.out.println("Noemployeesfound.");
                    }else{
                        System.out.println("\nEmployeeList:");
                        for (Employee emp : employees) {
                            System.out.println(emp);
                        }
                    }
                    break;
```

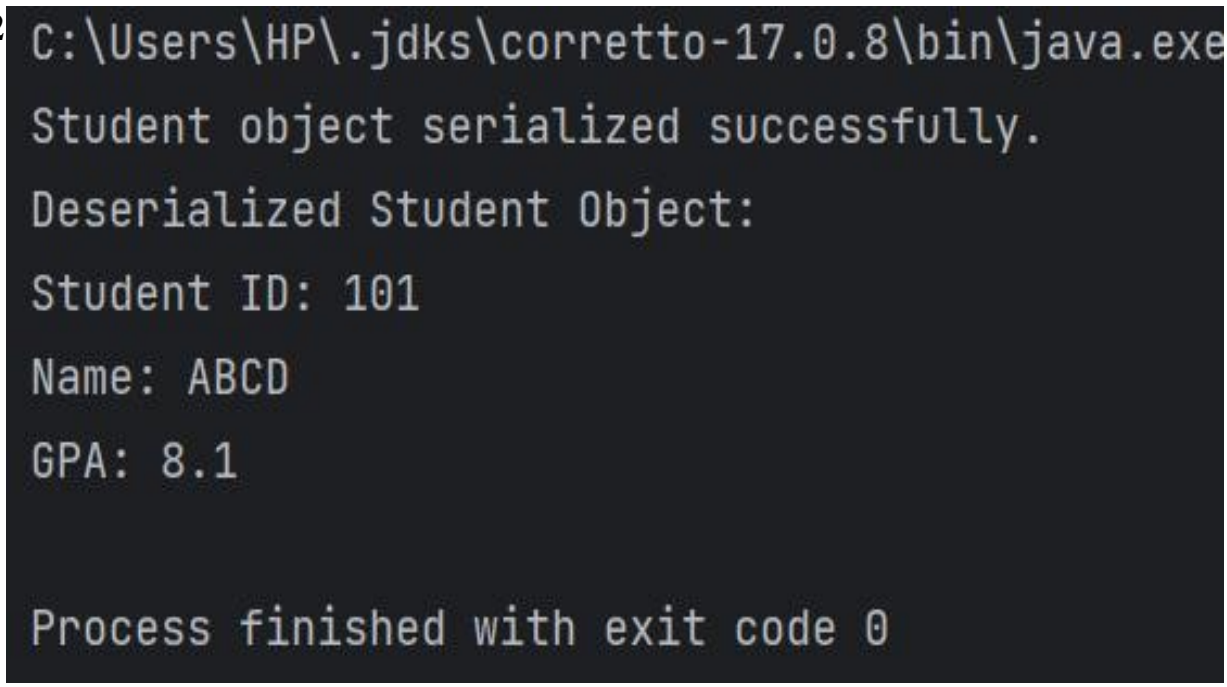
```
                case3:
                    System.out.println("Exitingapplication.");
                    scanner.close();
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalidchoice.Pleasetryagain.");
            }
        }
    }
}
```

4. Output:

4.1



4.2



4.3

```
1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter Employee ID: 101
Enter Name: ABCD
Enter Designation: Manager
Enter Salary: 110000
Employee added successfully.

1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 2

Employee List:
ID: 101, Name: ABCD, Designation: Manager, Salary: 110000.0
```

5. LearningOutcomes:

- UnderstandautoboxingandunboxinginJava.
- Learnobjectserializationanddeserializationusingstreams.
- HandleexceptionslikeIOExceptionandClassNotFoundException.
- Workwithcollectionsandperformarithmeticoperations.
- Usetry-with-resourcesforefficientfilehandling.
- Implementamenu-drivenemployeemanagementsystemusingcollections.