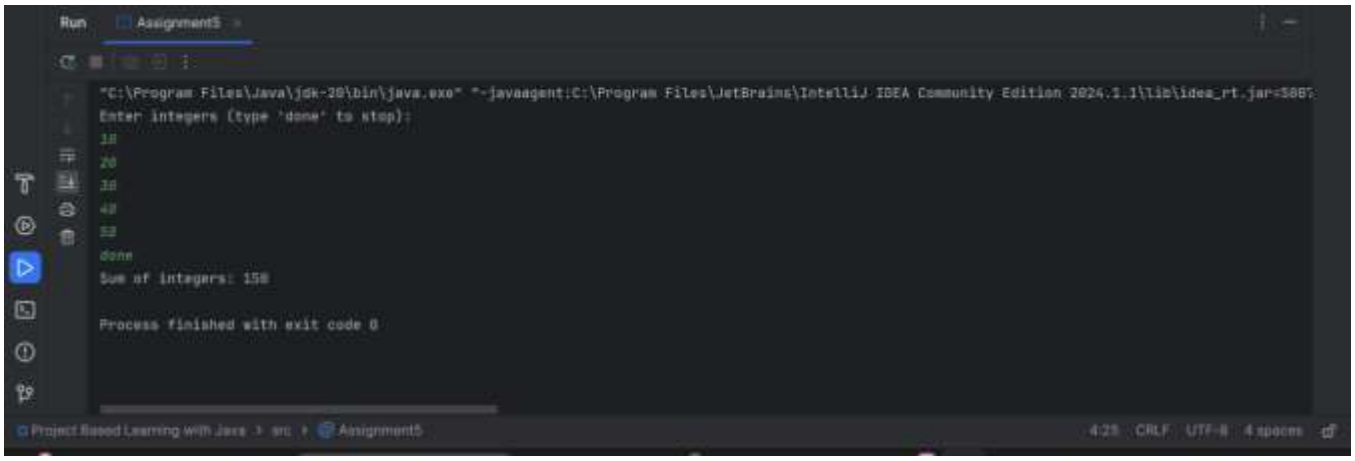
NAME : Agrima Sharma **UID :** 22BCS15314 **SECTION :** 607-A

Easy Level: Sum of Integers Using Autoboxing and Unboxing

```
import java.util.ArrayList;
import java.util.Scanner;

public class AutoboxingExample {
    public static int sumOfIntegers(ArrayList<Integer> numbers) {
        int sum = 0;
        for (Integer num : numbers) { // Unboxing (Integer → int)
            sum += num;
        }
        return sum;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Integer> numbers = new ArrayList<>();

        System.out.println("Enter integers (type 'done' to stop):");
        while (scanner.hasNext()) {
            String input = scanner.next();
            if (input.equalsIgnoreCase("done")) break;
            try {
                numbers.add(Integer.parseInt(input)); // Autoboxing (int → Integer)
            } catch (NumberFormatException e) {
                System.out.println("Invalid input! Please enter an integer.");
            }
        }

        int sum = sumOfIntegers(numbers);
        System.out.println("Sum of integers: " + sum);

        scanner.close();
    }
}
```



Medium Level: Serialization and Deserialization of a Student Object

```
import java.io.*;

// Serializable 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("ID: " + id + ", Name: " + name + ", GPA: " + gpa);
    }
}

public class StudentSerialization {
    private static final String FILE_NAME = "student.ser";

    public static void serializeStudent(Student student) {
        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
            out.writeObject(student);
            System.out.println("Student object serialized successfully.");
        } catch (IOException e) {
```

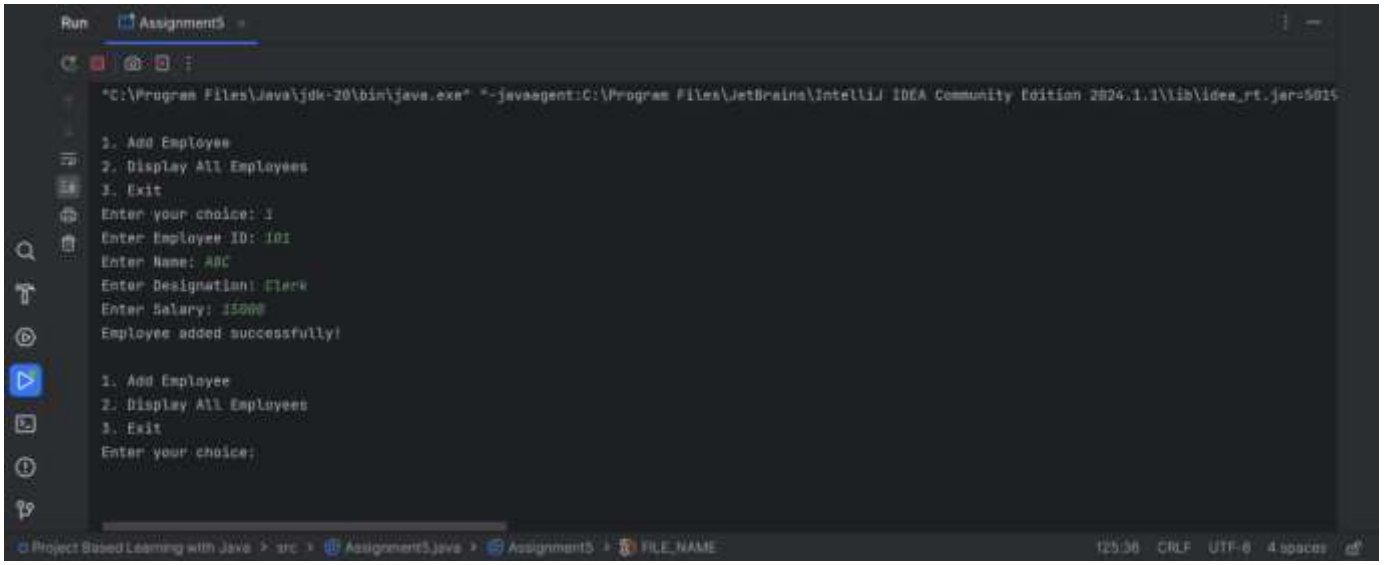
```
            System.out.println("Error during serialization: " + e.getMessage());
        }
    }

    public static void deserializeStudent() {
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            Student student = (Student) in.readObject();
            System.out.println("Deserialized  Student:");
            student.display();
        }    catch    (FileNotFoundException    e)    {
            System.out.println("Error: File not found.");
        } catch (IOException e) {
            System.out.println("Error during deserialization: " + e.getMessage());
        }    catch    (ClassNotFoundException    e)    {
            System.out.println("Error: Class not found.");
        }
    }

    public static void main(String[] args) {
        Student student = new Student(101, "Alice", 3.8);
        serializeStudent(student);
        deserializeStudent();
    }
}
```

Hard Level: Menu-Based Employee Management System

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

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("ID: " + id + ", Name: " + name + ", Designation: " +
designation + ", Salary: " + salary);
    }
}

public class EmployeeManagement {
    private static final String FILE_NAME = "employees.dat";
    private static Scanner scanner = new Scanner(System.in);

    public static void addEmployee() {
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        System.out.print("Enter Name: ");
        String name = scanner.nextLine();

        System.out.print("Enter Designation: ");
        String designation = scanner.nextLine();

        System.out.print("Enter Salary: ");
        double salary = scanner.nextDouble();

        Employee employee = new Employee(id, name, 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.out.println("Error while saving employee: " + e.getMessage());
        }
    }

    public static void displayAllEmployees() {
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            System.out.println("Employee List:");
            while (true) {
                Employee employee = (Employee) in.readObject();
                employee.display();
            }
        } catch (EOFException e) {
            // End of file reached
        } catch (FileNotFoundException e) {
            System.out.println("No employees found.");
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Error reading employees: " + e.getMessage());
        }
    }

    public static void main(String[] args) {
        while (true) {
            System.out.println("\n1. Add Employee\n2. Display All Employees\n3. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1 -> addEmployee();
                case 2 -> displayAllEmployees();
                case 3 -> {
                    System.out.println("Exiting...");
                    return;
                }
                default -> System.out.println("Invalid choice! Try again.");
            }
        }
    }
}
```


Run
Assignment5
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.1\lib\idea_rt.jar=5015
1. Add Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter Employee ID: 101
Enter Name: ABC
Enter Designation: Clerk
Enter Salary: 15000
Employee added successfully!
1. Add Employee
2. Display All Employees
3. Exit
Enter your choice:
Project Based Learning with Java > src > Assignment5.java > Assignment5 > FILE_NAME
125:36 CRLF UTF-8 4 spaces