

# EXPERIMENT 2.2

NAME	Taimur Kamran
UID	23BCS10578
CLASS	622-A

## PART – A

### Code:

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

public class SumWithAutoboxing {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter numbers (comma-separated): ");
        String input = sc.nextLine();
        String[] numbers = input.split(",");
        ArrayList<Integer> numList = new ArrayList<>();
        for (String numStr : numbers) {
            int number = Integer.parseInt(numStr.trim());
            numList.add(number);
        }
        int sum = 0;
```

```
for (Integer num : numList) {  
    sum += num;  
}  
System.out.println("Sum of numbers = " + sum);  
sc.close();  
}  
}
```

Output:

```
Output  
Enter numbers (comma-separated): 10,20,30,  
Sum of numbers = 60  
  
=== Code Execution Successful ===
```

## PART-B

**Code:**

```
import java.io.*;  
  
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 int getId() { return id; }

public String getName() { return name; }

public double getGpa() { return gpa; }

}

public class StudentSerializationDemo {

    public static void main(String[] args) {

        String filename = "student.ser";

        Student student = new Student(101, "Alice", 9.1);

        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(filename))) {

            out.writeObject(student);

            System.out.println("Student serialized successfully!");

        } catch (FileNotFoundException e) {

            System.out.println("Error: File not found.");

        } catch (IOException e) {

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

        }

        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(filename))) {

            Student deserializedStudent = (Student) in.readObject();

```

```
        System.out.println("\nStudent deserialized:");

        System.out.println("ID: " + deserializedStudent.getId());

        System.out.println("Name: " + deserializedStudent.getName());

        System.out.println("GPA: " + deserializedStudent.getGpa());
    } 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: Student class not found.");
    }
}
}
```

## Output:

yaml

student serialized successfully!

Student deserialized:

ID: 101

Name: Alice

GPA: 9.1

# PART-C

## Code:

```
import java.io.*;

import java.util.Scanner;

class Employee {

    private String name;

    private String id;

    private String designation;

    private double salary;

    public Employee(String name, String id, String designation, double salary) {

        this.name = name;

        this.id = id;

        this.designation = designation;

        this.salary = salary;

    }

    @Override

    public String toString() {

        return name + " | " + id + " | " + designation + " | " + salary;

    }

    public String toFileString() {

        return name + "," + id + "," + designation + "," + salary;

    }

    public static Employee fromFileString(String line) {

        String[] parts = line.split(",");

        return new Employee(parts[0], parts[1], parts[2], Double.parseDouble(parts[3]));

    }

}
```

```
}
```

```
}
```

```
public class EmployeeMenuApp {  
    private static final String FILE_NAME = "employees.txt";  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int choice;  
        while (true) {  
            System.out.println("\nMenu:");  
            System.out.println("1. Add Employee");  
            System.out.println("2. Display All");  
            System.out.println("3. Exit");  
            System.out.print("\nEnter choice: ");  
            choice = sc.nextInt();  
            sc.nextLine(); // consume newline  
            switch (choice) {  
                case 1:  
                    addEmployee(sc);  
                    break;  
                case 2:  
                    displayAllEmployees();  
                    break;  
                case 3:  
                    System.out.println("Exiting program...");  
                    sc.close();  
                    System.exit(0);  
                default:  
                    System.out.println("Invalid choice! Please try again.");  
            }  
        }  
    }  
}
```

```

    }
}
}

private static void addEmployee(Scanner sc) {
    System.out.print("Name: ");
    String name = sc.nextLine();
    System.out.print("ID: ");
    String id = sc.nextLine();
    System.out.print("Designation: ");
    String designation = sc.nextLine();
    System.out.print("Salary: ");
    double salary = sc.nextDouble();
    sc.nextLine(); // consume newline
    Employee emp = new Employee(name, id, designation, salary);
    try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE_NAME, true))) {
        bw.write(emp.toFileString());
        bw.newLine();
        System.out.println("Employee added successfully!");
    } catch (IOException e) {
        System.out.println("Error writing to file: " + e.getMessage());
    }
}

private static void displayAllEmployees() {
    System.out.println("\nEmployee List:");
    try (BufferedReader br = new BufferedReader(new FileReader(FILE_NAME))) {
        String line;
        while ((line = br.readLine()) != null) {
            Employee emp = Employee.fromFileString(line);

```

```
        System.out.println(emp);
    }
} catch (FileNotFoundException e) {
    System.out.println("No employee records found.");
} catch (IOException e) {
    System.out.println("Error reading from file: " + e.getMessage());
}
}
}
```

## Output:

```
Menu:
1. Add Employee
2. Display All
3. Exit

Enter choice: 1
Name: John
ID: 1001
Designation: Manager
Salary: 75000
Employee added successfully!

Enter choice: 2

Employee List:
John | 1001 | Manager | 75000

Enter choice: 3
Exiting program...
```