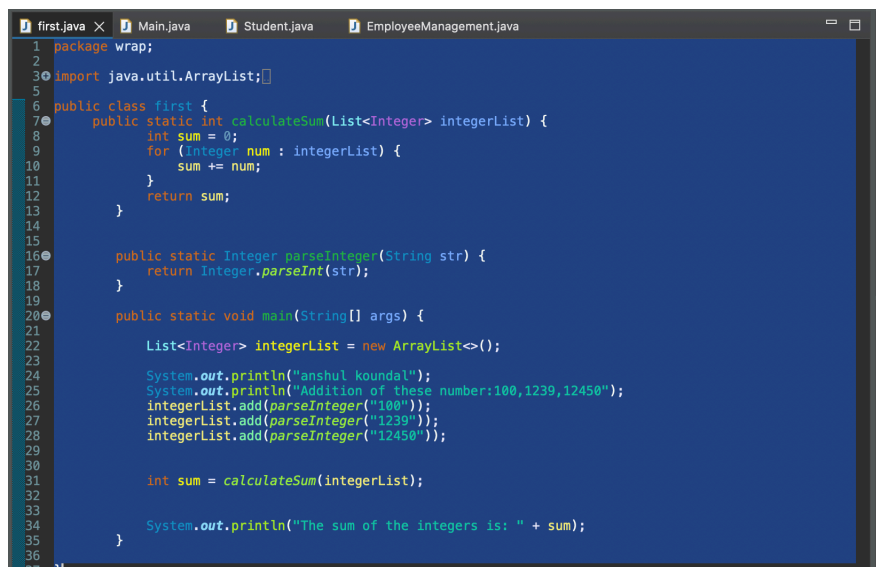


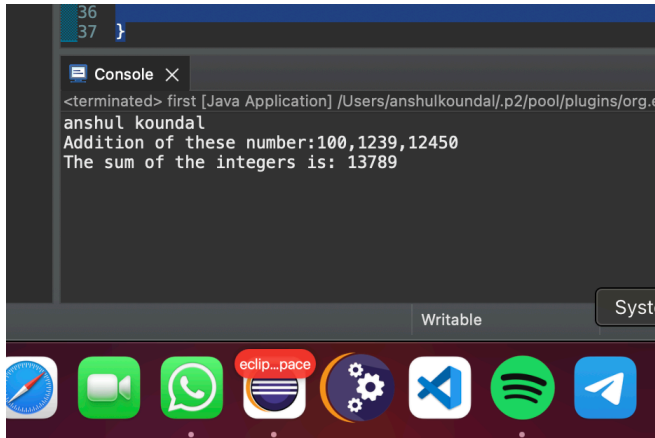
**Name :-Anshul koundal**  
**UID:-22BCS16818**  
**CLASS:-22BCS-IOT-619 B**

## #CODE

```
package wrap;
import java.util.ArrayList;
import java.util.List;
public class first {
    public static int calculateSum(List<Integer> integerList) {
        int sum = 0;
        for (Integer num : integerList) {
            sum += num;
        }
        return sum;
    }
    public static Integer parseInt(String str) {
        return Integer.parseInt(str);
    }
    public static void main(String[] args) {
        List<Integer> integerList = new ArrayList<>();
        System.out.println("anshul koundal");
        System.out.println("Addition of these number:100,1239,12450");
        integerList.add(parseInt("100"));
        integerList.add(parseInt("1239"));
        integerList.add(parseInt("12450"));
        int sum = calculateSum(integerList);
        System.out.println("The sum of the integers is: " + sum);
    }
}
```



The screenshot shows an IDE window with the following tabs: first.java, Main.java, Student.java, and EmployeeManagement.java. The code in first.java is displayed with line numbers 1 through 37. The code is identical to the one provided in the previous block, showing the package declaration, imports, class definition, and the main method that calculates the sum of integers in a list.



## #code 2

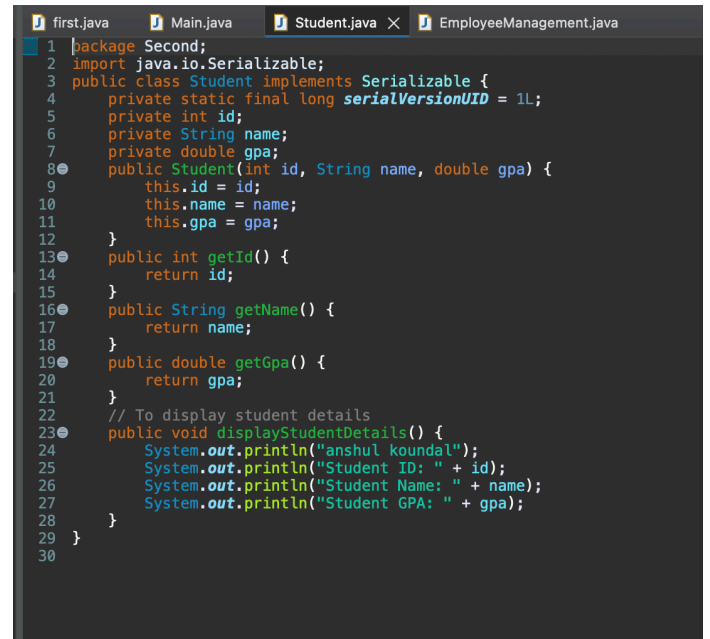
### —main.java—

```
package Second;
import java.io.*;
public class Main {
    public static void serializeStudent(Student student, String filename) {
        try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(filename))) {
            out.writeObject(student);
            System.out.println("Student object serialized successfully.");
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("IOException occurred: " + e.getMessage());
        }
    }
    public static Student deserializeStudent(String filename) {
        Student student = null;
        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(filename))) {
            student = (Student) in.readObject(); // Deserializes the student object
            System.out.println("Student object deserialized successfully.");
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("IOException occurred: " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.out.println("Class not found: " + e.getMessage());
        }
        return student;
    }
    public static void main(String[] args) {

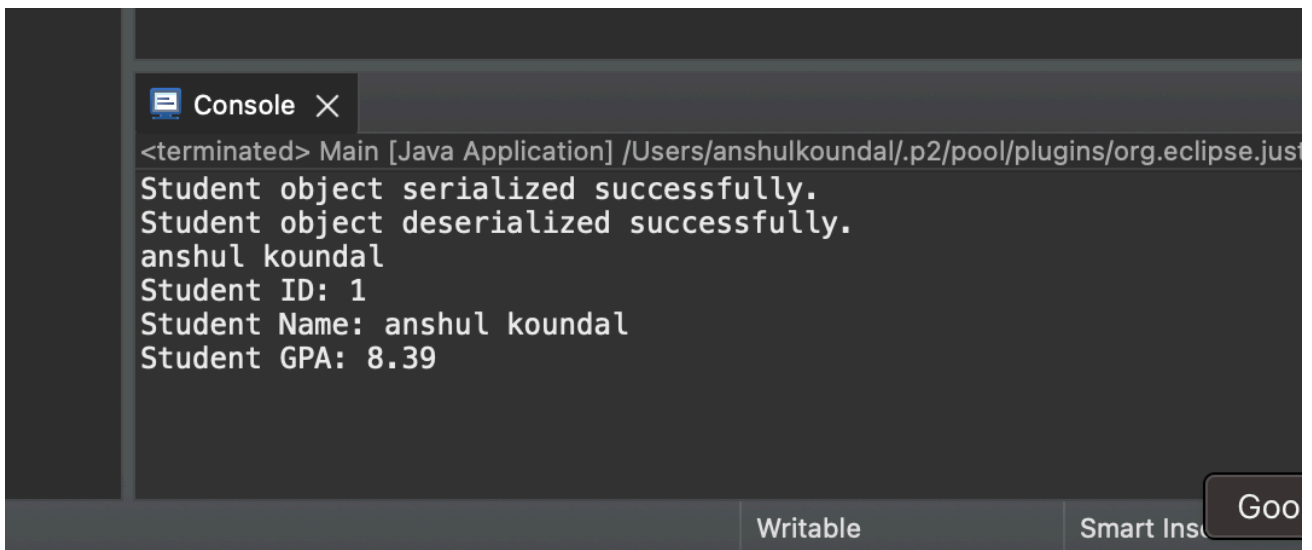
        Student student = new Student(1, "anshul koundal ", 8.39);
        String filename = "student.ser";
        serializeStudent(student, filename);
        Student deserializedStudent = deserializeStudent(filename);
        if (deserializedStudent != null) {
            deserializedStudent.displayStudentDetails();
        }
    }
}
```

## Student.java

```
package Second;
import java.io.Serializable;
public 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;
    }
    // To display student details
    public void displayStudentDetails() {
        System.out.println("anshul koundal");
        System.out.println("Student ID: " + id);
        System.out.println("Student Name: " + name);
        System.out.println("Student GPA: " + gpa);
    }
}
```



```
1 package Second;
2 import java.io.Serializable;
3 public class Student implements Serializable {
4     private static final long serialVersionUID = 1L;
5     private int id;
6     private String name;
7     private double gpa;
8     public Student(int id, String name, double gpa) {
9         this.id = id;
10        this.name = name;
11        this.gpa = gpa;
12    }
13    public int getId() {
14        return id;
15    }
16    public String getName() {
17        return name;
18    }
19    public double getGpa() {
20        return gpa;
21    }
22    // To display student details
23    public void displayStudentDetails() {
24        System.out.println("anshul koundal");
25        System.out.println("Student ID: " + id);
26        System.out.println("Student Name: " + name);
27        System.out.println("Student GPA: " + gpa);
28    }
29 }
30
```



```
<terminated> Main [Java Application] /Users/anshulkoundal/.p2/pool/plugins/org.eclipse.jst
Student object serialized successfully.
Student object deserialized successfully.
anshul koundal
Student ID: 1
Student Name: anshul koundal
Student GPA: 8.39
```

## #code 3

```
package Third;
import java.sql.*;
import java.util.Scanner;

public class EmployeeManagement {

    private static Connection connection;
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
private static Scanner scanner = new Scanner(System.in);

private static void connectToDatabase() {
    try {

        String url = "jdbc:mysql://localhost:3306/employee_db";
        String username = "root";
        String password = "Kapil@8009";

        connection = DriverManager.getConnection(url, username, password);
        System.out.println("Connected to the database.");
    } catch (SQLException e) {

        System.out.println("Database connection failed: " + e.getMessage());
    }
}

private static void addEmployee() {
    try {

        if (connection == null) {
            System.out.println("Database connection is not established.");
            return;
        }
        System.out.print("Enter Employee Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Designation: ");
        String designation = scanner.nextLine();
        System.out.print("Enter Salary: ");
        double salary = scanner.nextDouble();
        scanner.nextLine();
        String query = "INSERT INTO employees1 (employee_id, employee_name, designation, salary) VALUES
(? , ? , ? , ?)";
        PreparedStatement stmt = connection.prepareStatement(query);
        System.out.println("anshul koundal");
        stmt.setInt(1, id);
        stmt.setString(2, name);
        stmt.setString(3, designation);
        stmt.setDouble(4, salary);

        int result = stmt.executeUpdate();
        if (result > 0) {
            System.out.println("Employee added successfully!");
        } else {
            System.out.println("Failed to add employee.");
        }
    } catch (SQLException e) {

        System.out.println("Error adding employee: " + e.getMessage());
    }
}

private static void displayAllEmployees() {
    try {

        if (connection == null) {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        System.out.println("Database connection is not established.");
        return;
    }
    String query = "SELECT * FROM employees1";
    Statement stmt = connection.createStatement();
    ResultSet rs = stmt.executeQuery(query);
    while (rs.next()) {
        System.out.println("Employee ID: " + rs.getInt("employee_id"));
        System.out.println("Employee Name: " + rs.getString("employee_name"));
        System.out.println("Designation: " + rs.getString("designation"));
        System.out.println("Salary: " + rs.getDouble("salary"));
        System.out.println("-----");
    }

    } catch (SQLException e) {

        System.out.println("Error displaying employees: " + e.getMessage());
    }
}
private static void closeConnection() {
    try {
        if (connection != null) {
            connection.close();
            System.out.println("Database connection closed.");
        }
    } catch (SQLException e) {

        System.out.println("Error closing connection: " + e.getMessage());
    }
}
public static void main(String[] args) {

    connectToDatabase();
    while (true) {
        System.out.println("\nMenu:");
        System.out.println("1. Add an Employee");
        System.out.println("2. Display All Employees");
        System.out.println("3. Exit");
        System.out.print("Select an option: ");
        int choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
            case 1:
                addEmployee();
                break;
            case 2:
                displayAllEmployees();
                break;
            case 3:
                closeConnection();
                System.out.println("Exiting application.");
                return;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
first.java Main.java Student.java *EmployeeManagement.java X
1 package Third;
2 import java.sql.*;
3 import java.util.Scanner;
4
5 public class EmployeeManagement {
6
7     private static Connection connection;
8     private static Scanner scanner = new Scanner(System.in);
9
10    private static void connectToDatabase() {
11        try {
12            String url = "jdbc:mysql://localhost:3306/employee_db";
13            String username = "root";
14            String password = "Kapil@0009";
15
16            connection = DriverManager.getConnection(url, username, password);
17            System.out.println("Connected to the database.");
18        } catch (SQLException e) {
19            System.out.println("Database connection failed: " + e.getMessage());
20        }
21    }
22
23    private static void addEmployee() {
24        try {
25            if (connection == null) {
26                System.out.println("Database connection is not established.");
27                return;
28            }
29            System.out.print("Enter Employee Name: ");
30            String name = scanner.nextLine();
31            System.out.print("Enter Employee ID: ");
32            int id = scanner.nextInt();
33            scanner.nextLine();
34            System.out.print("Enter Designation: ");
35            String designation = scanner.nextLine();
36            System.out.print("Enter Salary: ");
37            double salary = scanner.nextDouble();
38            scanner.nextLine();
39            String query = "INSERT INTO employees1 (employee_id, employee_name, designation, salary) VALUES (?, ?, ?, ?)";
40            PreparedStatement stmt = connection.prepareStatement(query);
41            System.out.println("anshul koundal");
42            stmt.setInt(1, id);
43            stmt.setString(2, name);
44            stmt.setString(3, designation);
45            stmt.setDouble(4, salary);
46
47            int result = stmt.executeUpdate();
48            if (result > 0) {
49                System.out.println("Employee added successfully!");
50            }
51        } catch (SQLException e) {
52            System.out.println("Error adding employee: " + e.getMessage());
53        }
54    }
55
56    private static void displayAllEmployees() {
57        try {
58            if (connection == null) {
59                System.out.println("Database connection is not established.");
60                return;
61            }
62            String query = "SELECT * FROM employees1";
63            Statement stmt = connection.createStatement();
64            ResultSet rs = stmt.executeQuery(query);
65            while (rs.next()) {
66                System.out.println("Employee ID: " + rs.getInt("employee_id"));
67                System.out.println("Employee Name: " + rs.getString("employee_name"));
68                System.out.println("Designation: " + rs.getString("designation"));
69                System.out.println("Salary: " + rs.getDouble("salary"));
70            }
71        } catch (SQLException e) {
72            System.out.println("Error displaying employees: " + e.getMessage());
73        }
74    }
75
76    private static void closeConnection() {
77        try {
78            if (connection != null) {
79                connection.close();
80                System.out.println("Database connection closed.");
81            }
82        } catch (SQLException e) {
83            System.out.println("Error closing connection: " + e.getMessage());
84        }
85    }
86
87    public static void main(String[] args) {
88        connectToDatabase();
89        while (true) {
90            System.out.println("\nMenu:");
91            System.out.println("1. Add an Employee");
92            System.out.println("2. Display All Employees");
93            System.out.println("3. Exit");
94            System.out.print("Select an option: ");
95            int choice = scanner.nextInt();
96            scanner.nextLine();
97
98            switch (choice) {
99                case 1:
100                    addEmployee();
101                    break;
102                case 2:
103                    displayAllEmployees();
104                    break;
105                case 3:
106                    closeConnection();
107                    System.out.println("Exiting application.");
108                    return;
109                default:
110                    System.out.println("Invalid choice. Please try again.");
111            }
112        }
113    }
114 }
```

```
0
1 switch (choice) {
2     case 1:
3         addEmployee();
4         break;
5     case 2:
6         displayAllEmployees();
7         break;
8     case 3:
9         closeConnection();
10        System.out.println("Exiting application.");
11        return;
12    default:
13        System.out.println("Invalid choice. Please try again.");
14    }
15 }
16 }
17 }
```

## Output

```
126 }
127 }

Console X
<terminated> EmployeeManagement [Java Application] /Users/anshulkoun
Database connection failed: No suitable driver found f

Menu:
1. Add an Employee
2. Display All Employees
3. Exit
Select an option: 3
Exiting application.
```