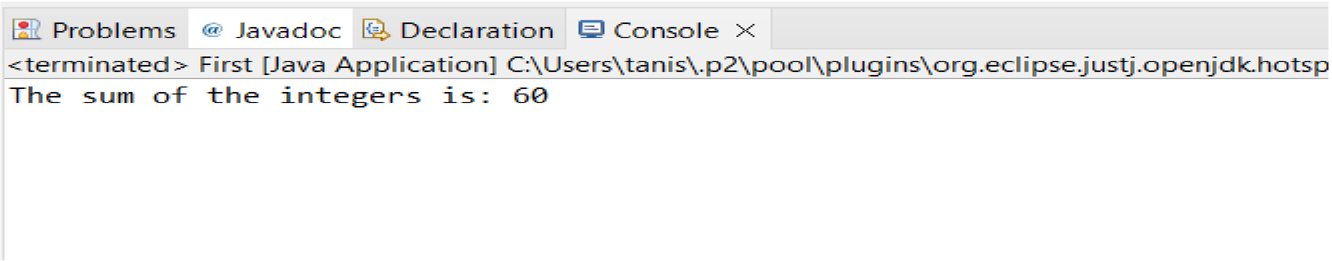
WORKSHEET 05

Name :Tanishk Rautela
UID: 22BCS16821
SECTION: 22BCS_IOT-619
GROUP: A

#CODE 1

```
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 parseInteger(String str) {
        return Integer.parseInt(str);
    }
    public static void main(String[] args) {
        List<Integer> integerList = new ArrayList<>();
        integerList.add(parseInteger("10"));
        integerList.add(parseInteger("20"));
        integerList.add(parseInteger("30"));
        int sum = calculateSum(integerList);
        System.out.println("The sum of the integers is: " + sum);
    }
}
```

OUTPUT



```
<terminated> First [Java Application] C:\Users\tanis\.p2\pool\plugins\org.eclipse.justj.openjdk.hotsp
The sum of the integers is: 60
```

#CODE 2

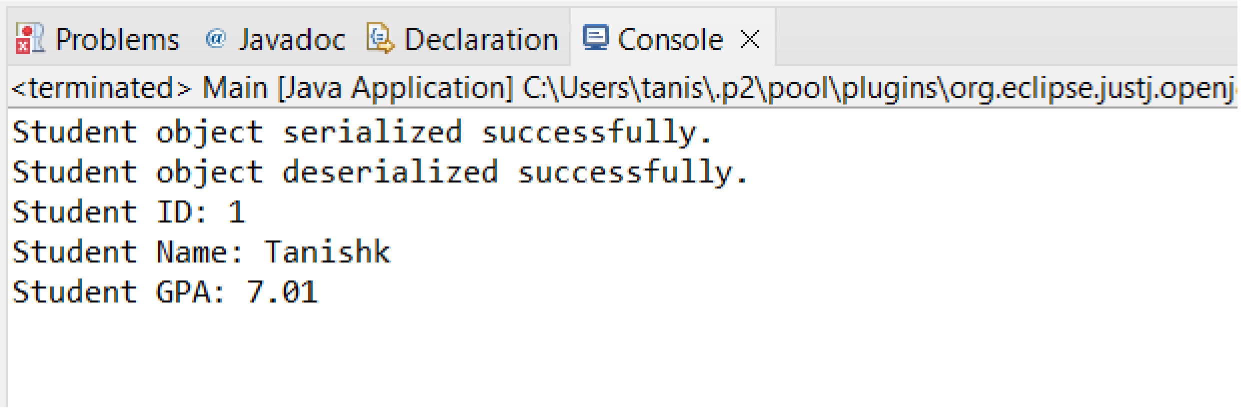
———————————-main.java————————————

```
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, "Tanishk", 7.01);


        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("Student ID: " + id);
        System.out.println("Student Name: " + name);
        System.out.println("Student GPA: " + gpa);
    }
}
```

OUTPUT

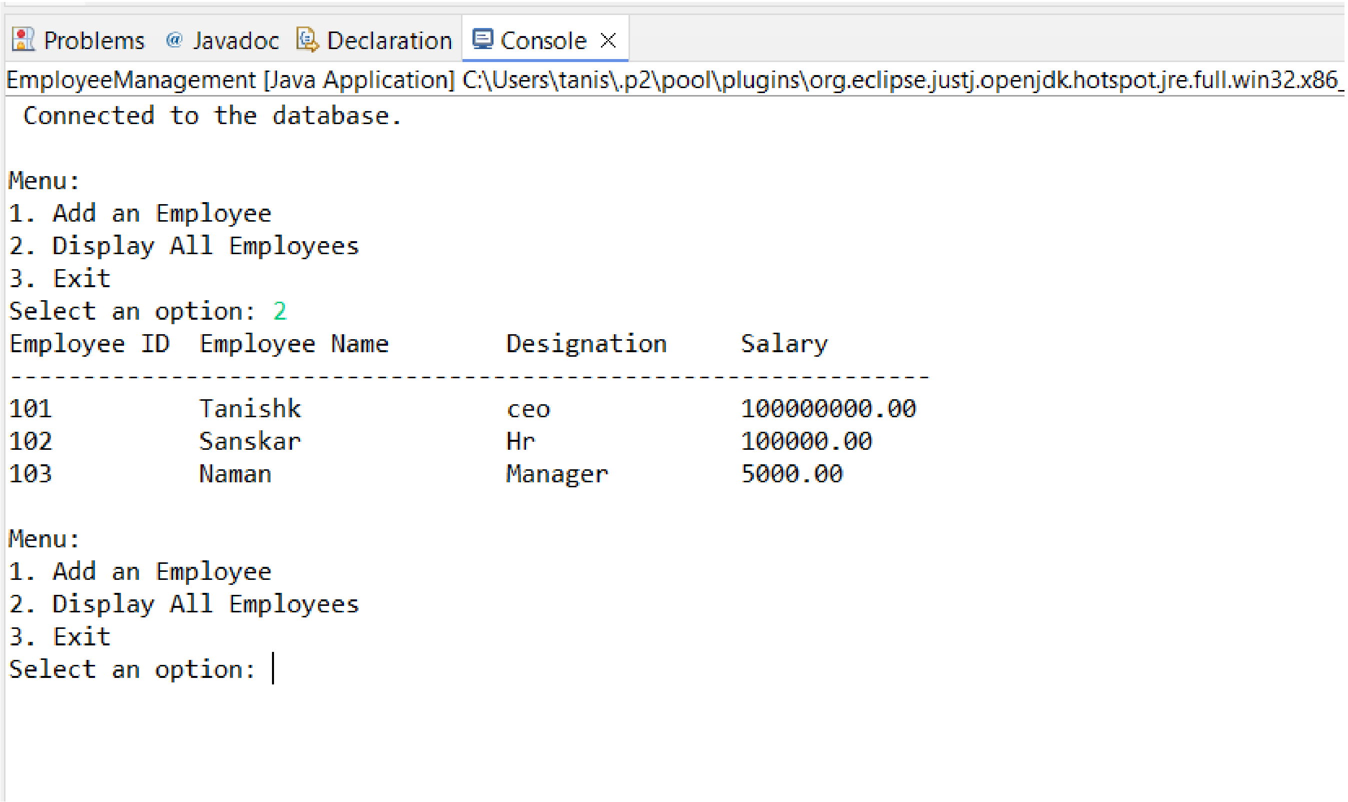


```
<terminated> Main [Java Application] C:\Users\tanis\.p2\pool\plugins\org.eclipse.justj.openj
Student object serialized successfully.
Student object deserialized successfully.
Student ID: 1
Student Name: Tanishk
Student GPA: 7.01
```

#CODE 3

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

public class EmployeeManagement {

    private static Connection connection;
    private static final Scanner scanner = new Scanner(System.in);

    private static void connectToDatabase() {
        try {
            String url = "jdbc:mysql://localhost:3306/sys";
            String username = "root";
            String password = "Tanishk@1602";

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

    private static boolean employeeExists(int id) {
        try {
            String checkQuery = "SELECT 1 FROM employees1 WHERE employee_id = ?";
            PreparedStatement checkStmt = connection.prepareStatement(checkQuery);
            checkStmt.setInt(1, id);
            ResultSet rs = checkStmt.executeQuery();
            return rs.next(); // Returns true if employee exists
        } catch (SQLException e) {
            System.out.println(" Error checking employee: " + e.getMessage());
            return false;
        }
    }

    private static void addEmployee() {
        try {
            if (connection == null) {
                System.out.println(" Database connection is not established.");
                return;
            }

            System.out.print("Enter Employee ID: ");
            int id = scanner.nextInt();
            scanner.nextLine();

            if (employeeExists(id)) {
                System.out.println(" Employee with ID " + id + " already exists.");
                return;
            }

            System.out.print("Enter Employee Name: ");
            String name = 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);
            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());
            e.printStackTrace();
        }
    }

    private static void displayAllEmployees() {
        try {
            if (connection == null) {
                System.out.println(" Database connection is not established.");
                return;
            }

            String query = "SELECT * FROM employees1";
            Statement stmt = connection.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            System.out.printf("%-12s %-20s %-15s %-10s\n", "Employee ID", "Employee Name", "Designation", "Salary");
            System.out.println("------------------------------------------------------------");

            boolean hasEmployees = false;
            while (rs.next()) {
                hasEmployees = true;
                System.out.printf("%-12d %-20s %-15s %.2f\n",
                        rs.getInt("employee_id"),
                        rs.getString("employee_name"),
                        rs.getString("designation"),
                        rs.getDouble("salary"));
            }

            if (!hasEmployees) {
                System.out.println(" No employees found.");
            }

        } catch (SQLException e) {
            System.out.println(" Error displaying employees: " + e.getMessage());
            e.printStackTrace();
        }
    }

    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());
            e.printStackTrace();
        }
    }

    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: ");

            if (!scanner.hasNextInt()) {
                System.out.println("Invalid input. Please enter a valid option.");
                scanner.next();
                continue;
            }

            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.");
            }
        }
    }
}
```

OUTPUT



EmployeeManagement [Java Application] C:\Users\tanis\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_

```
 Connected to the database.

Menu:
1. Add an Employee
2. Display All Employees
3. Exit
Select an option: 2
Employee ID  Employee Name        Designation     Salary
--------------------------------------------------------------
101          Tanishk              ceo             100000000.00
102          Sanskar              Hr              100000.00
103          Naman                Manager         5000.00

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