Soumyajit Mondal - 23262325

TELECOM NMS - JAVA

DAY 1

Generic Class

```java
package Java_AdvanceTraining_WIPRO;

public class Day1_Generic_Class {

    int id;
    String name;
    String dept;
    String designation;

    public Day1_Generic_Class (int id, String name, String dept,
            String designation) {
        this.id = id;
        this.name = name;
        this.dept = dept;
        this.designation = designation;
    }

    @Override
    public String toString() {
        return "Day1_Generic_Class[id =" + id + ", name =" + name +","
                + " dept =" + dept + ", designation =" + designation +"]";
    }
}

class Generics1 <X>{
    X data;
    void assign (X data) {
        this.data = data;
}

    void display() {
        System.out.println("Given data : " + data);
    }

```

```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Generics1 <Integer> intObj = new Generics1 <Integer>();
        intObj.assign(100);
        intObj.display();

        Generics1 <String> StrObj = new Generics1 <String>();
        StrObj.assign("Java Generics");
        StrObj.display();

        Day1_Generic_Class empObj = new Day1_Generic_Class (101,
                "Soumyajit", "IT", "Full Stack Developer");
        Generics1<Day1_Generic_Class> empGen =
                new Generics1<Day1_Generic_Class>();
        empGen.assign(empObj);
        empGen.display();

    }
}

```

Problems Javadoc Declaration Console

```
<terminated> Generics1 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (19-Apr-2024, 2:41:47 pm – 2:41:47 pm) [pid: 4696]
Given data : 100
Given data : Java Generics
Given data : Day1_Generic_Class[id =101, name =Soumyajit, dept =IT, designation = Full Stack Developer]
```

Generic Methods

```java
package Java_AdvanceTraining_WIPRO;

public class Day1_Generic_Methods {

    public static <E> void printArray (E[] inputArray) {

        System.out.println();
        for (E el: inputArray)
            System.out.print(el + ",");
    }

```

```
11
12      public static void main(String[] args) {
13          // TODO Auto-generated method stub
14
15          Integer[] intArray = { 1,2,3,4,5};
16          Double[] doubleArray = {1.1, 1.2, 1.3, 1.4, 1.5};
17          Character[] charArray = {'j','a','v','a'};
18
19          printArray(intArray);
20          printArray(doubleArray);
21          printArray(charArray);
22      }
23
24  }
25
```

```
Problems  Javadoc  Declaration  Console ×
<terminated> Day1_Generic_Methods [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (19-Apr-2024, 2:46:28 pm – 2:46:28 pm) [pid: 2864]
1,2,3,4,5,
1.1,1.2,1.3,1.4,1.5,
j,a,v,a,
```


Java Reflection

Reflection Class Information

```
1  package Java_AdvanceTraining_WIPRO;
2
3  import java.lang.reflect.Modifier;
4
5  class Employee2{
6
7  }
8
```

```java
public class Day1_Reflection_Class {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Employee2 emp = new Employee2();
        Class obj =  emp.getClass();

        String className = obj.getName();
        System.out.println("Name of the class is : " +className);

        int intModifiers = obj.getModifiers();

        String strMod = Modifier.toString(intModifiers);
        System.out.println("Modifiers of the class : "+strMod);

        Class superClass = obj.getSuperclass();
        System.out.println("Super class of Employee2 is : "+
            superClass.getName());
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> Day1_Reflection_Class [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (19-Apr-2024, 2:51:43 pm – 2:51:43 pm) [pid: 7916]
Name of the class is : Java_AdvanceTraining_WIPRO.Employee2
Modifiers of the class :
Super class of Employee2 is : java.lang.Object
```


Reflection Method Information

```java
package Java_AdvanceTraining_WIPRO;

import java.lang.reflect.Method;

class Employee3 {
        public String getData() {
        return "Returning the data";
    }

    public void setData() {

    }

}

public class Day1_Refelction_Method {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

```

```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Employee3 emp = new Employee3();
        Class obj = emp.getClass();

        Method listOfMethods[] = obj.getDeclaredMethods();
        for (Method m: listOfMethods) {
            String methodName = m.getName();
            System.out.println("\nMethod name is :" +methodName);

            int intModifier = m.getModifiers();
            String strModifier = Modifier.toString(intModifier);
            System.out.println("Modifier of the class : " + strModifier);
            System.out.println("Method return type : " + m.getReturnType());
        }

    }

}
```

Line: 33

Task List | Problems | Javadoc | Declaration | Console

<terminated> Day1_Refelction_Method [Java Application] C:\Program Files\Java\jdk-21\b

```
Method name is :setData
Modifier of the class : public
Method return type : void

Method name is :getData
Modifier of the class : public
Method return type : class java.lang.String
```

Reflection Accessing Private Attribute

```java
package Java_AdvanceTraining_WIPRO;

import java.lang.reflect.Field;

class Employee4{
    private String name;
}


public class Day1_Reflection_PrivateAttribute {
```

```java
public class Day1_Reflection_PrivateAttribute {

    public static void main(String[] args) throws NoSuchFieldException,
    SecurityException, IllegalArgumentException, IllegalAccessException {
        // TODO Auto-generated method stub

        Employee4 emp = new Employee4();
        Class obj = emp.getClass();
        Field field1 = obj.getDeclaredField("name");
        field1.setAccessible(true);
        field1.set(emp, "Soumya");

        String nameData = (String) field1.get(emp);
        System.out.println("\nPrivate data modified...Name value : "
         +nameData);
        int modifier1 = field1.getModifiers();
        String strModifier1 = Modifier.toString(modifier1);
        System.out.println("Field Modifier is : "+ strModifier1);

    }

}
```

Task List | Problems | Javadoc | Declaration | Console

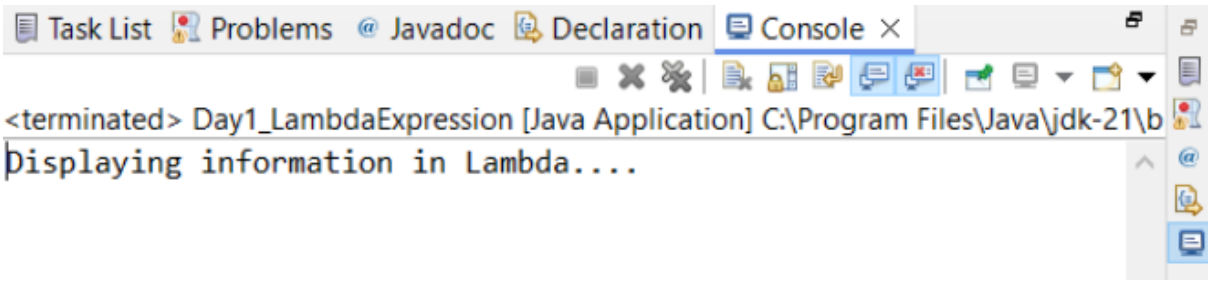
<terminated> Day1_Reflection_PrivateAttribute [Java Application] C:\Program Files\Java\j

```
Private data modified...Name value : Soumya
Field Modifier is : private
```

Lambda Expression

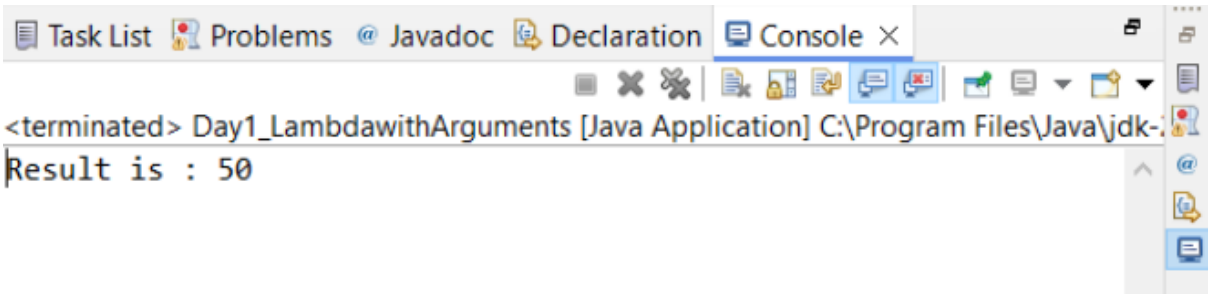
Lambda Expression

```java
package Java_AdvanceTraining_WIPRO;

interface ITest{
    public void display();
}

public class Day1_LambdaExpression {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        ITest t1 = () -> System.out.println("Displaying information "
                + "in Lambda....");
        t1.display();

    }

}
```



Lambda With Arguments

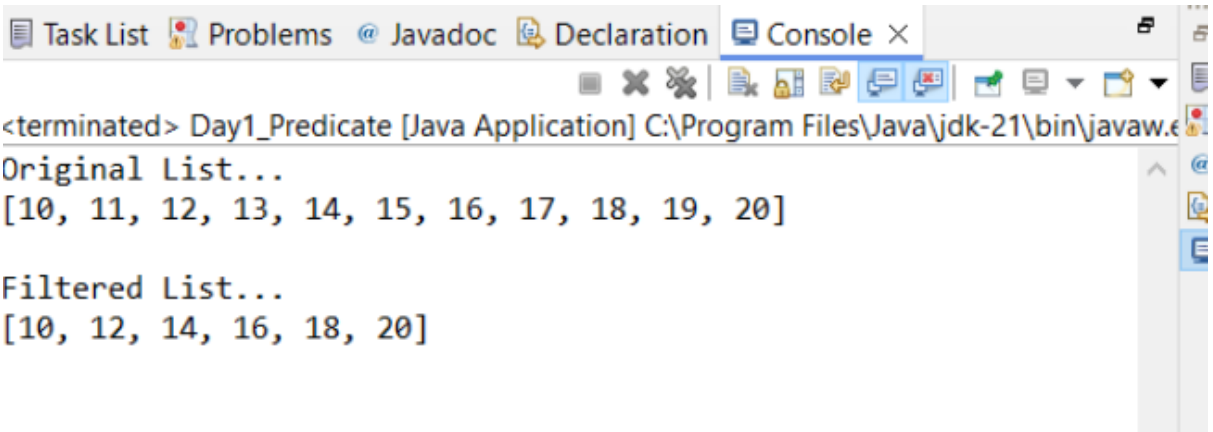
```java
package Java_AdvanceTraining_WIPRO;

interface IMaths{
    public int sum(int x, int y);
}

public class Day1_LambdawithArguments {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        IMaths m1 = (x,y) -> x + y;
        int x  = 20 , y = 30;
        int result = m1.sum(x, y);
        System.out.println("Result is : " + result);

    }

}
```



Predicate

Predicate

```java
package Java_AdvanceTraining_WIPRO;

import java.util.Arrays;

public class Day1_Predicate {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        List<Integer> numLst = Arrays.asList(10, 11, 12, 13, 14, 15,
                16, 17, 18, 19, 20);
        System.out.println("Original List...");
        System.out.println(numLst);

        Predicate<Integer> p1 = data -> data %2 == 0;

        List filteredList = numLst.
                            stream().
                            filter(p1).
                            collect(Collectors.toList());
        System.out.println("\nFiltered List...");
        System.out.println(filteredList);
    }

}
```



Method Referencing

```java
package Java_AdvanceTraining_WIPRO;

interface INewMap{
    public void displayDirections();
}

public class Day1_Method_Referencing {

    Day1_Method_Referencing(){
        System.out.println("Constructor of the class Test...");
    }
    public void displayGoogleMap() {
    System.out.println("Displaying google map from src to destination..");
    }
    public static void displayGoogleMapsStatic() {
    System.out.println("Displaying google maps from a static method..");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Day1_Method_Referencing Day1_Method_ReferencingObj =
                new Day1_Method_Referencing();
        INewMap IN = Day1_Method_ReferencingObj :: displayGoogleMap;
        IN.displayDirections();

    INewMap InStatic = Day1_Method_Referencing::displayGoogleMapsStatic;
    InStatic.displayDirections();

        INewMap INCons = Day1_Method_Referencing :: new;
        INCons.displayDirections();

    }

}
```

Task List | Problems | Javadoc | Declaration | Console

```
<terminated> Day1_Method_Referencing [Java Application] C:\Program Files\Java\jdk-21\
Constructor of the class Test...
Displaying google map from src to destination..
Displaying google maps from a static method..
Constructor of the class Test...
```

DAY 2

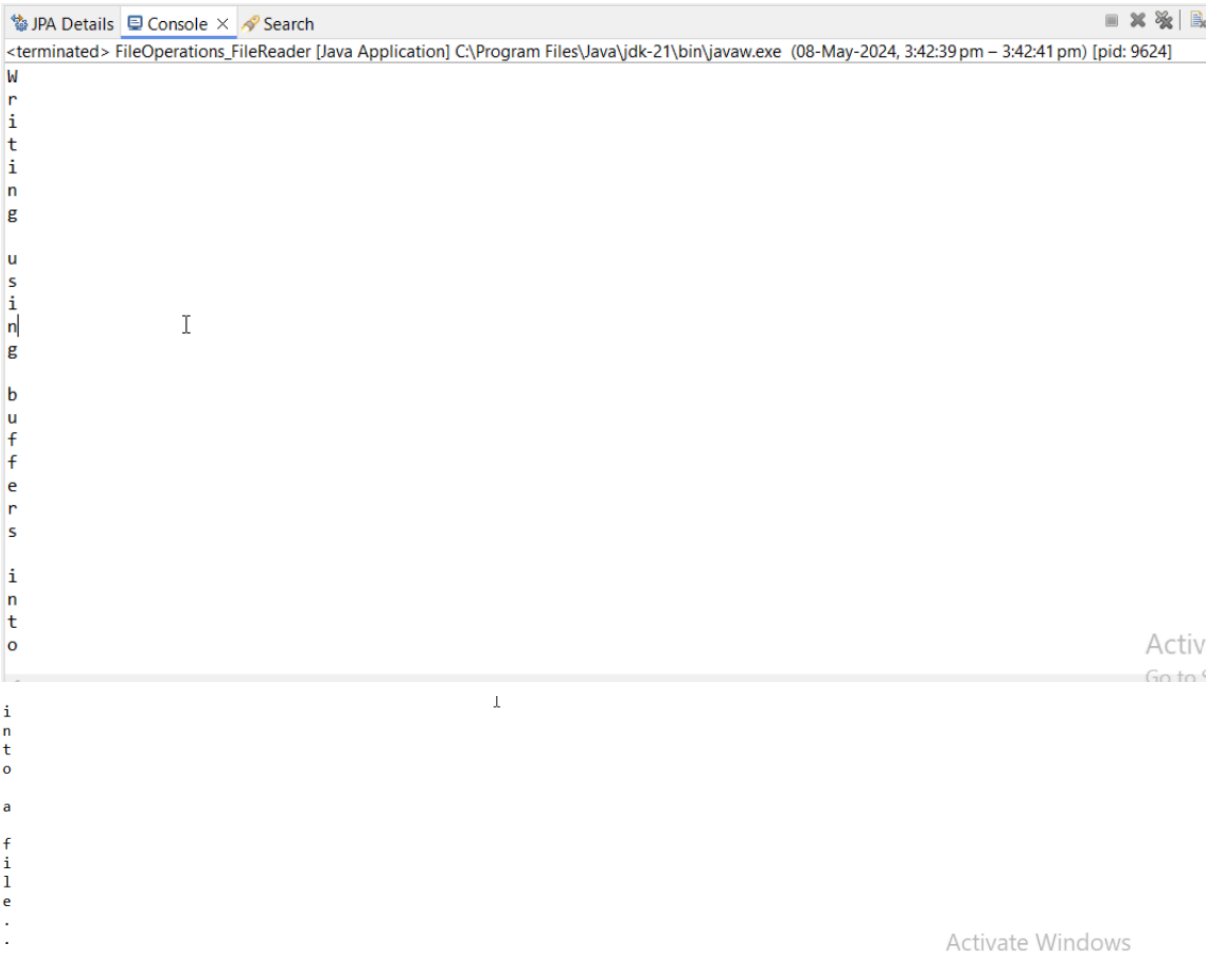
File Operations

FileWriter

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.FileWriter;

public class FileOperations_FileWriter {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub

        FileWriter fw = new FileWriter("C:\\TELECOM NMS JAVA");

        fw.write("hello, welcome to java..");
        System.out.println("Data writting into a file successfully..");

        fw.close();
    }

}
```

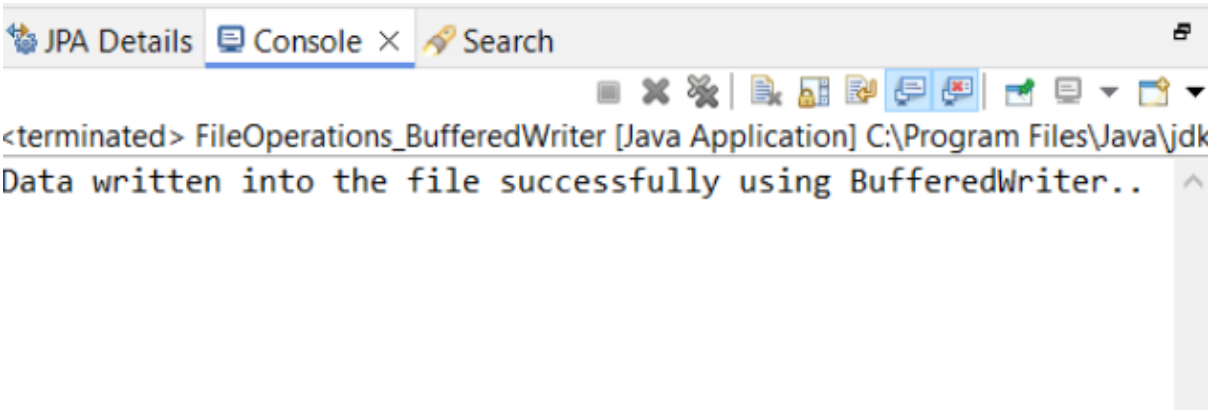
FileReader

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.FileNotFoundException;

public class FileOperations_FileReader {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {
            FileReader fr = new FileReader ("C:\\\\TELECOM NMS JAVA\\\\FileWriter.JAVA");

            int i = 0;
            while ((i = fr.read()) != -1) {
                System.out.println((char)i);
            }
            fr.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();

        } catch (IOException e) {
            e.printStackTrace();
        }

    }

}
```



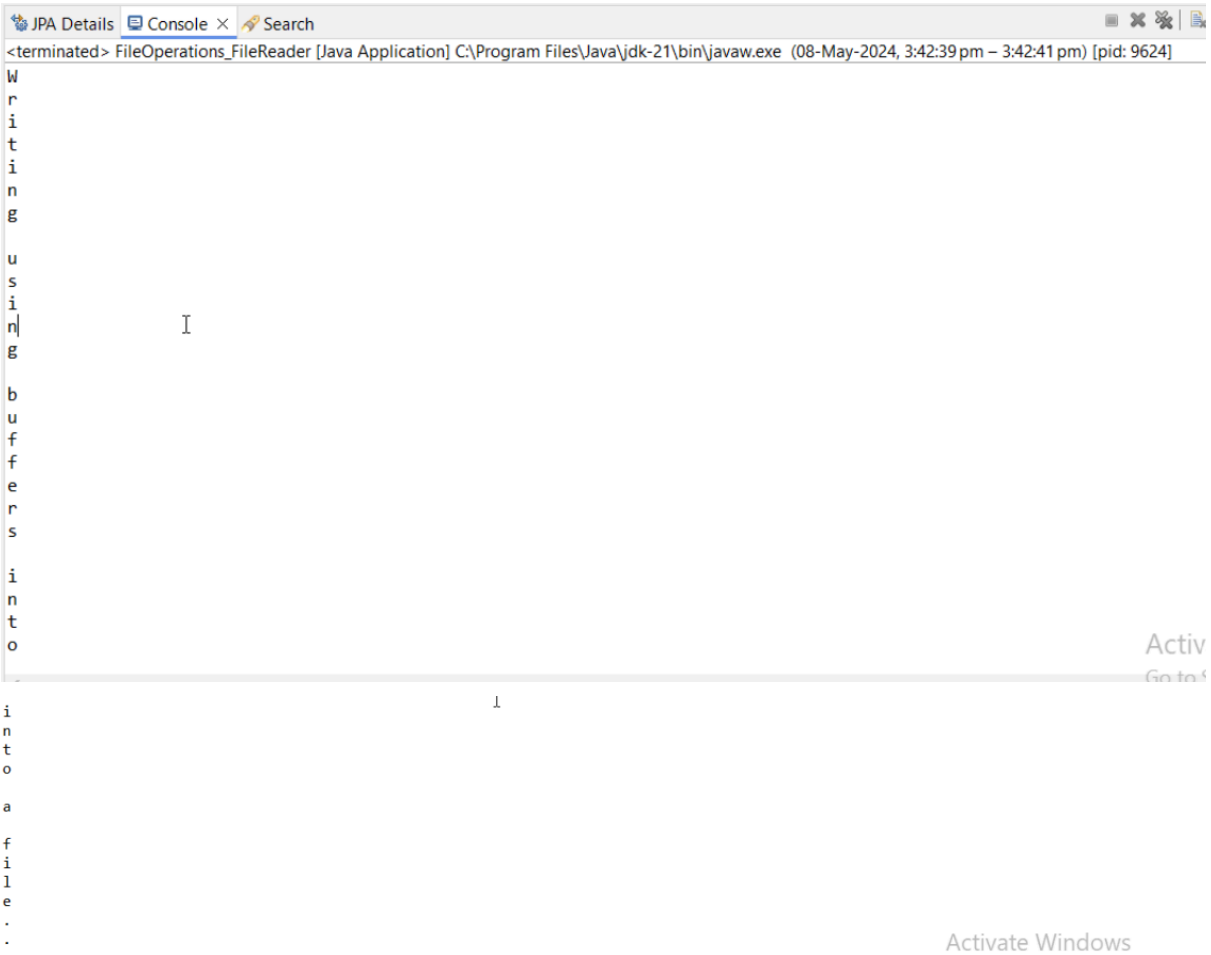
BufferedWriter

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.BufferedWriter;

public class FileOperations_BufferedWriter {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {
            FileWriter fw = new FileWriter("C:\\\\\\\\TELECOM NMS JAVA\\\\\\\\FileWriter.JAVA");
            BufferedWriter buffer  = new BufferedWriter (fw);
            buffer.write("Writing using buffers into a file..");
            System.out.println("Data written into the file successfully using BufferedWriter..");
            buffer.close();

        } catch (IOException e) {
            e.printStackTrace();
        }

    }

}
```



BufferedReader

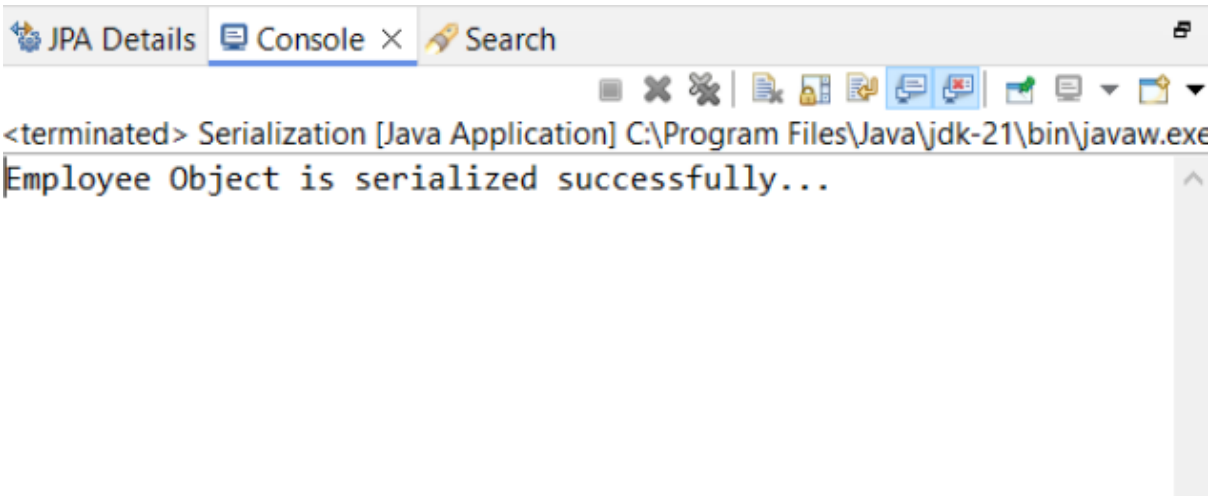
```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.BufferedReader;

public class FileOperations_BufferedReader {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {
            FileReader fr = new FileReader("C:\\\\\\\\\TELECOM NMS JAVA\\\\\\\\\FileWriter.JAVA");
        BufferedReader br = new BufferedReader(fr);
        int i = 0;
        while((i = br.read()) != -1) {
            System.out.println((char)i);
        }
        } catch (FileNotFoundException e) {
            e.printStackTrace();

        } catch (IOException ex) {
            System.out.println("Exception is :" + ex);
        }

    }

}
```



Serialization

```
1  package Java_AdvanceTraining_Day2_Wipro;
2
3  import java.io.FileNotFoundException;
8
9  class EmployeeSerialization implements Serializable{
10
11     int id;
12     String name;
13     String dept;
14     String designation;
15
16 EmployeeSerialization (int id, String name, String dept,
17         String designation) {
18     this.id = id;
19     this.name = name;
20     this.dept = dept;
21     this.designation = designation;
22    }
23
24 public String toString() {
25 return "Employee Serialization [id=" + id + ", name=" + name + ", "
26         + "dept=" + dept + ","
27         + " designation =" + designation + "]";
28 }
29 }
30
```

```java
public class Serialization {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        EmployeeSerialization empObj = new EmployeeSerialization
                (101, "Soumyajit", "IT", "Full Stack Developer");
        try {
        FileOutputStream fos = new FileOutputStream
                ("C:SoumyaPractice.txt");
        ObjectOutputStream oos = new ObjectOutputStream (fos);
        oos.writeObject (empObj);

    System.out.println("Employee Object is serialized successfully...");
        fos.close();
        oos.close();

    }catch (FileNotFoundException ex) {
            System.out.println(ex);


    }catch (IOException ex) {
            System.out.println(ex);
        }
    }

}

```

Deserialization

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.FileInputStream;

class EmployeeSerialization implements Serializable{
    int id;
    String name;
    String dept;
    String designation;

EmployeeSerialization (int id, String name, String dept,
        String designation) {
    this.id = id;
    this.name = name;
    this.dept = dept;
    this.designation = designation;
   }

public String toString() {
return "Employee Serialization [id=" + id + ", name=" + name + ", "
        + "dept=" + dept + ","
        + " designation =" + designation + "]";
    }
}

public class Deserialization {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        EmployeeSerialization empObj = new EmployeeSerialization
                (101, "Soumyajit", "IT", "Full Stack Developer");
        try {
        FileInputStream fis = new FileInputStream
                ("C:SoumyaPractice.txt");

        ObjectInputStream ois = new ObjectInputStream (fis);
        EmployeeSerialization empobj =
                (EmployeeSerialization) ois.readObject();

        System.out.println("Object deserialized is: " + empobj);

        }catch (FileNotFoundException ex) {
            System.out.println(ex);

        } catch (IOException ex) {
          System.out.println(ex);

        } catch (ClassNotFoundException ex) {
          System.out.println(ex);

        }

    }
}
```



```
JPA Details  Console ×  Search
<terminated> Deserialization [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (08-May-2024, 3:52:17 pm – 3:52:20 pm) [pid: 9244]
Object deserialized is: Employee Serialization [id=101, name=Soumyajit, dept=IT, designation =Full Stack Developer]
```

Directory Operations

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.File;

public class Directory_Operations {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String directoryPath = "C:SoumyaPractice.txt";

        //Defining the directory path.
        File directory = new File (directoryPath);

        //Get all the files from a directory and store in files array.

        File files[] = directory.listFiles();


       //Display all the files in the given directory.
     System.out.println("List of file in the directory: "+
         directoryPath);

        if (files!= null) {
         for (File file: files)
             //For each loop.. for (<data type> <Object N System.out.println(file.getName());
             System.out.println(file.getName());
    }

}
}
```

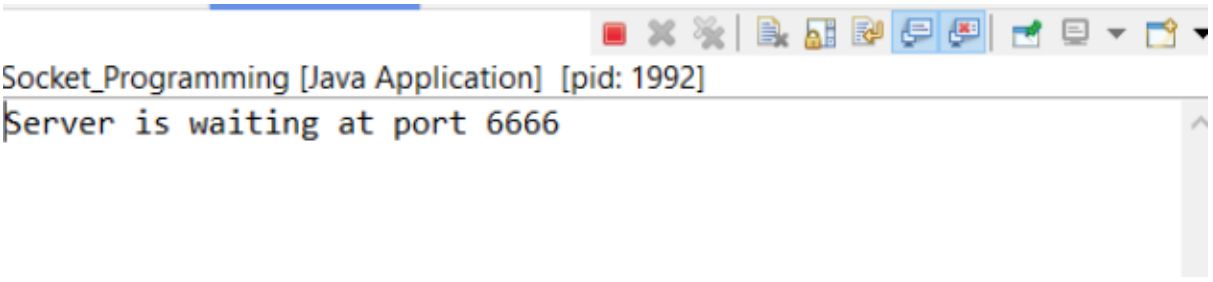


```
<terminated> Directory_Operations [Java Application] C:\Program Files\Java\jdk-21\bin\j
List of file in the directory: C:SoumyaPractice.txt
```

Socket Programming

Server Socket

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.io.DataInputStream;

public class Socket_Programming {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {

        ServerSocket ss = new ServerSocket (6666);
        System.out.println("Server is waiting at port 6666");

        Socket s = ss.accept(); //Establlishes a connection.


            //Connected with the client data..
    DataInputStream dis = new DataInputStream (s.getInputStream());
    String str = dis.readUTF(); //Reads the client data..
    System.out.println("Received data from client is: "+str);
          ss.close();
    } catch (IOException ex) {
                System.out.println(ex);

    }

}
}
```

Client Socket

```
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;

public class ClientSocket {
    public static void main(String args[]) {
        try {
            //Connecting to server running at localhost at port 6666.
            Socket s = new Socket ("localhost", 6666);

            //Data output stream is connected to Socket
            DataOutputStream dos = new DataOutputStream (s.getOutputStream());

            dos.writeUTF("Hello Java...");

            dos.flush(); //Clears the buffer.

            dos.close();


        }catch (IOException ex) {
            System.out.println(ex);
        }
    }

}
```

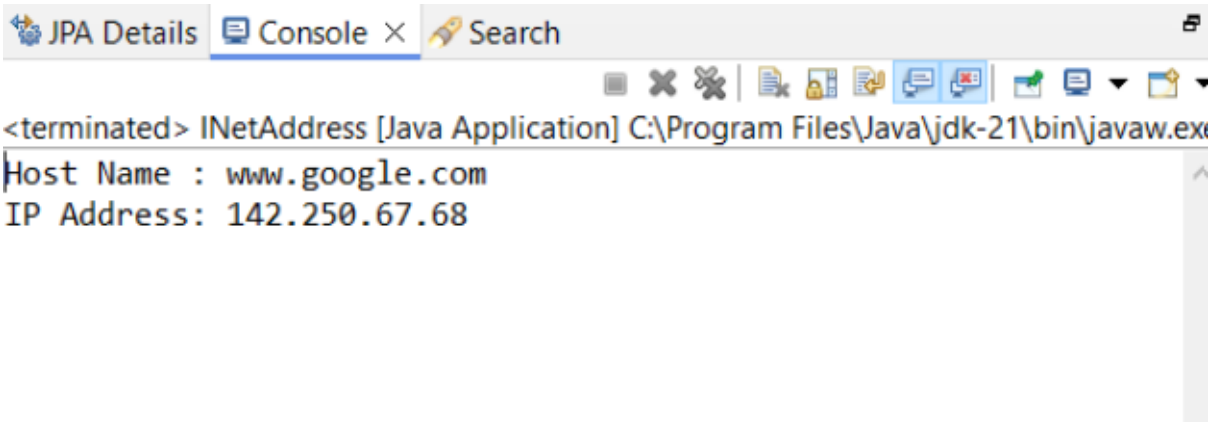
```
Server is waiting at port 6666
Received data from client is :Hello Java...
```

URL

```
package Wipro.Day2;

import java.io.File;
import java.net.MalformedURLException;
import java.net.URL;

public class Test{

    public static void main (String args[]) {

        URL url;
        try {
            url = new URL ("http://www.google.com");
            System.out.println("Protocol : "+url.getProtocol());
            System.out.println("Host name : " + url.getHost());


        } catch (MalformedURLException e) {

            e.printStackTrace();
        }
```

INet Address

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.net.InetAddress;

public class INetAddress {
    public static void main (String args[]) {

        try {

            InetAddress IA = InetAddress.getByName ("www.google.com");
            System.out.println("Host Name : " + IA.getHostName());
            System.out.println("IP Address: "+ IA.getHostAddress());

        } catch (UnknownHostException e) {
            e.printStackTrace();
        }

    }

}
```



Local Date , Time

```java
package Java_AdvanceTraining_Day2_Wipro;

import java.time.LocalDate;

public class LocalDateTime {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        LocalDate ld = LocalDate.now();

        System.out.println("Todays date is: "+1d);

        int day = ld.getDayOfMonth();

        int month = ld.getMonthValue();

        int year = ld.getYear();

        System.out.println("Today's date is: "+day+ "-" + month + "-"+year);

        LocalTime lt = LocalTime.now();

        System.out.println("Current time is: " + lt);

```

```
25
26          int hour = lt.getHour();
27
28          int min = lt.getMinute();
29
30          int second = lt.getSecond();
31
32          System.out.println("Current time is : " + hour + ":" + min + ":" + second);
33
34
35
36      }
37
38
39  }
```


JPA Details | Console | Search

<terminated> LocalDateTime [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.

```
Todays date is: 1.0
Today's date is: 8-5-2024
Current time is: 16:31:42.931644200
Current time is : 16:31:42
```

TimeZones and ZoneIDs

```
3  import java.time.LocalTime;
4  import java.time.ZoneId;
5  import java.util.Set;
6
7  public class Test{
8
9      public static void main (String args[]) {
0
1        LocalTime lt = LocalTime.now(ZoneId.of("Singapore"));
2        System.out.println("Current time in Singapore : "+lt);
3
4        Set<String> zoneSet = ZoneId.getAvailableZoneIds();
5        System.out.println("List of Zones are : ");
6
7         for (String zone : zoneSet)
8            System.out.println(zone);
9      }
0
1  }
```

```
List of Zones are :
Asia/Aden
America/Cuiaba
Etc/GMT+9
Etc/GMT+8
Africa/Nairobi
America/Marigot
Asia/Aqtau
```

DAY 3

Driver Class

Insert-Update-Delete

```java
package Java_AdvanceTraining_Day3_Wipro;

import java.sql.Connection;

public class Driver_Class {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
        // Load the driver
        Class.forName("com.mysql.cj.jdbc.Driver");

        //Connect with the database
        Connection con = DriverManager.getConnection
                ("jdbc:mysql://127.0.0.1:3306/test", "root", "admin" );
        System.out.println("Connected to the database....");

        Statement st = con.createStatement();
st.executeUpdate("INSERT INTO Employee(name,  Employee_ID) VALUES('Soumyajit',40000)");
        con.close();
    }catch(ClassNotFoundException e) {

    }catch (SQLException ex) {
        System.out.println(ex);
    }

}
}
```



```
<terminated> Driver_Class [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe
Connected to the database....
```

Prepared Statement

```java
package Java_AdvanceTraining_Day3_Wipro;

import java.sql.Connection;

public class Prepared_Statement {

    private static String strQry;

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {
        //Load the driver.
        Class.forName ("com.mysql.cj.jdbc.Driver");

        //Connect with the database.
        Connection con = DriverManager.getConnection
                ("jdbc:mysql://127.0.0.1:3306/test", "root", "admin");
        System.out.println("Connected to the database...");

        String strory = "INSERT INTO emp (name, sal) VALUES (?,?)";
        PreparedStatement ps = con.prepareStatement(strQry);
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter a name: ");
        String strName = sc.next();
        System.out.println("Enter Salary: ");
        int sal = sc.nextInt();
        int sal = sc.nextInt();

        ps.setString(1, strName);

        ps.setInt(2, sal);

        ps.executeUpdate();

        System.out.println("Inserted Record in the db Successfully..");

        con.close();

        System.out.println("Closed the database connection successfully...");

        }catch (ClassNotFoundException e) {

        } catch (SQLException ex) {
            System.out.println(ex);
        }

        }

    }
```



```
Connected to the database...
Enter a name :
Kavya
Enter Salary :
35000
Inserted Record in the db Successfully..
Closed the database connection successfully...
```

ResultSet

```java
package Java_AdvanceTraining_Day3_Wipro;

import java.sql.Connection;

public class ResultSet {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {
//Load the driver.

Class.forName ("com.mysql.cj.jdbc.Driver");

//Connect with the database.

Connection con = DriverManager.getConnection
("jdbc:mysql://127.0.0.1:3306/test", "root", "admin");
                System.out.println("Connected to the database...");

Statement st = con.createStatement();

java.sql.ResultSet rs = st.executeQuery("SELECT * FROM emp");

System.out.println("List of records in emp table are: ");

 while (rs.next()) {

System.out.print(rs.getString(1));
System.out.println(" "+ rs.getInt(2));
con.close();
System.out.println("Closed the database connection successfully...");
 }}
        catch (ClassNotFoundException e) {

} catch (SQLException ex) {

System.out.println(ex);
    }
}
    public static String getInt(String string) {
        // TODO Auto-generated method stub
        return null;
    }

    public static String getString(String string) {
        // TODO Auto-generated method stub
        return null;
    }

    public static boolean next() {
        // TODO Auto-generated method stub
        return false;
    }
}
```

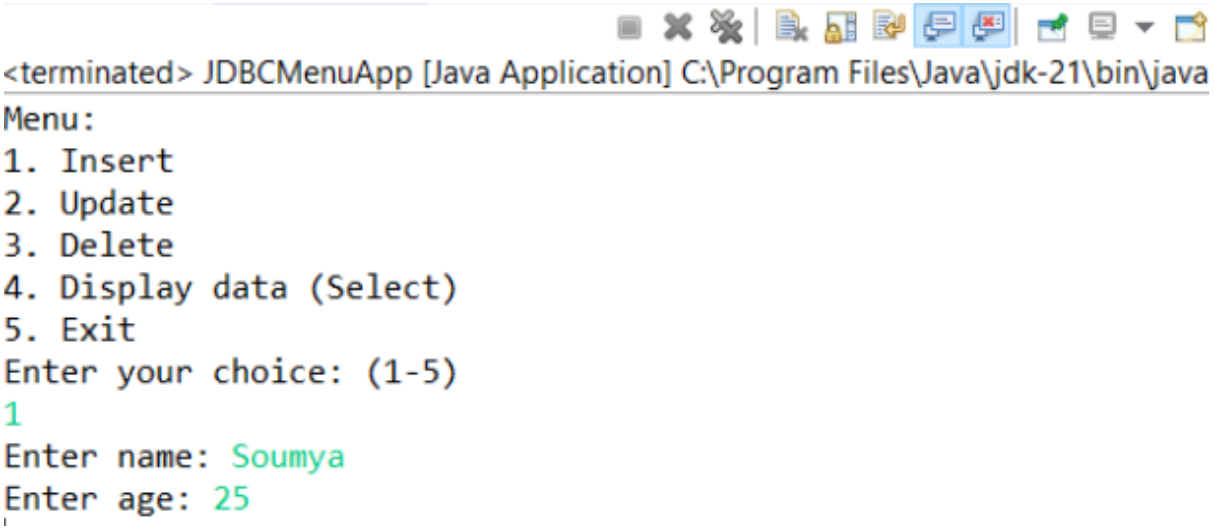


```
Connected to the database...
List of records in emp table are :
Charan.K 1000
Shiva 1000
Sai Rakesh 30000
Trupti 30000
Kavya 35000
Closed the database connection successfully...
```

JDBC MenuApp

```java
package Java_AdvanceTraining_Day3_Wipro;

import java.sql.Connection;

public class JDBCMenuApp {

    // Database connection details
    private static final String DB_URL = "jdbc:mysql://127.0.0.1:3306/test"; // Modify as needed
    private static final String DB_USER = "root"; // Modify as needed
    private static final String DB_PASSWORD = "admin"; // Modify as needed

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try (Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD)) {
            Scanner scanner = new Scanner(System.in);
            boolean exit = false;
            while (!exit) {
                System.out.println("Menu:");
                System.out.println("1. Insert");
                System.out.println("2. Update");
                System.out.println("3. Delete");
                System.out.println("4. Display data (Select)");
                System.out.println("5. Exit");
                System.out.print("Enter your choice: (1-5) ");

                int choice = scanner.nextInt();
                scanner.nextLine();
                switch (choice) {
                case 1:
                    insertRecord(conn, scanner);
                    break;
                case 2:
                    updateRecord(conn, scanner);
                    break;
                case 3:
                    deleteRecord(conn, scanner);
                    break;
                case 4:
                    displayRecords(conn);
                    break;
                case 5:
                    exit = true;
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice, please try again.");
                    break;
                }
            }

            scanner.close();
```

```
60      } catch (SQLException e) {
61          System.err.println("Database connection error: " + e.getMessage());
62      }
63  }
64      private static void insertRecord(Connection conn, Scanner scanner) throws SQLException {
65          System.out.print("Enter name: ");
66          String name = scanner.nextLine();
67
68          System.out.print("Enter age: ");
69          int age = scanner.nextInt();
70
71          String sql = "INSERT INTO your_table (name, age) VALUES (?, ?)";
72          try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
73              pstmt.setString(1, name);
74              pstmt.setInt(2, age);
75              pstmt.executeUpdate();
76              System.out.println("Record inserted successfully.");
77          }
78      }
79
80      private static void updateRecord(Connection conn, Scanner scanner) throws SQLException {
81          System.out.print("Enter record ID to update: ");
82          int id = scanner.nextInt();
83          scanner.nextLine(); // Consume newline
84
85          System.out.print("Enter new name: ");
86          String newName = scanner.nextLine();
87
88          System.out.print("Enter new age: ");
89          int newAge = scanner.nextInt();
90
91          String sql = "UPDATE your_table SET name = ?, age = ? WHERE id = ?";
92          try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
93              pstmt.setString(1, newName);
94              pstmt.setInt(2, newAge);
95              pstmt.setInt(3, id);
96              pstmt.executeUpdate();
97              System.out.println("Record updated successfully.");
98          }
99      }
100
101     private static void deleteRecord(Connection conn, Scanner scanner) throws SQLException {
102         System.out.print("Enter record ID to delete: ");
103         int id = scanner.nextInt();
104
105         String sql = "DELETE FROM your_table WHERE id = ?";
106         try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
107             pstmt.setInt(1, id);
108             pstmt.executeUpdate();
109             System.out.println("Record deleted successfully.");
110         }
111     }
111     }
112     private static void displayRecords(Connection conn) throws SQLException {
113         String sql = "SELECT * FROM your_table"; // Replace 'your_table' with your actual table name
114         try (Statement stmt = conn.createStatement();
115                 ResultSet rs = stmt.executeQuery(sql)) {
116             System.out.println("ID | Name | Age");
117             while (rs.next()) {
118                 System.out.println(rs.getInt("id") + " | " + rs.getString("name") + " | " +
119             rs.getInt("age"));
120             }
121
122     }
123
124 }
125 }
```



DataBase MetaData

```java
package Java_AdvanceTraining_Day3_Wipro;

import java.sql.SQLException;

public class Database_MetaData {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {

        //Load the driver.

        Class.forName ("com.mysql.cj.jdbc.Driver");

        //Connect with the database.

        Connection con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test",
                "root", "admin");
        System.out.println("Connected to the database...");

        DatabaseMetaData dbMeta = con.getMetaData();



        System.out.println("Name of the database: "+dbMeta.getDatabaseProductName());
        System.out.println("Name of the driver: "+dbMeta.getDriverName());
```

```
31
32          System.out.println("Driver version:" + dbMeta.getDriverVersion());
33
34          System.out.println("Table length name: "+dbMeta.getMaxTableNameLength());
35
36          System.out.println("Maximum number of columns in a table: "+dbMeta.getMaxColumnsInTable());
37
38          con.close();
39
40          System.out.println("Closed the database connection successfully...");
41
42          System.out.println("Closed the database connection successfully...");
43          } catch (ClassNotFoundException e) {
44
45          } catch (SQLException ex) {
46
47          System.out.println(ex);
48
49          }
50
51
52
53          }
```

```
<terminated> Database_MetaData [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (08-May-2024, 5:13:24 pm – 5:1
Connected to the database...
Name of the database: MySQL
Name of the driver: MySQL Connector/J
Driver version:mysql-connector-j-8.3.0 (Revision: 805f872a57875f311cb82487efcfb070411a3fa0)
Table length name: 64
Maximum number of columns in a table: 512
Closed the database connection successfully...
Closed the database connection successfully...
```


Junit

```
1  package Java_AdvanceTraining_Day3_Wipro;
2
3  import org.junit.Test;
4
5  public class Junit_Suites {
6
7          public int Junit_Suites(int x, int y) {
8              return x * y;
9          }
10
11         public void Junit_Suites() {
12             Junit_Suites testObj = new Junit_Suites();
13             int x =  20, y = 3, finalResult = 60;
14
15             int functionResult = testObj.Junit_Suites(x, y);
16             assertEquals (finalResult, functionResult);
17         }
18
19         private void assertEquals(int finalResult, int functionResult) {
20             // TODO Auto-generated method stub
21
22         }
23
24     }
```

Runs: 1/1 Errors: 0 Failures: 0

DAY 4

Design Pattern

Singleton Pattern

```java
package Java_AdvanceTraining_Day4_Wipro;

class Whatsapp{
    private static Whatsapp whatsappObject = null;
    private Whatsapp() {}
    public static Whatsapp getInstance() {
        if(whatsappObject == null)
            whatsappObject = new Whatsapp();
        return whatsappObject;
    }
}

public class Singleton {

    public static void main(String args []) {
        // TODO Auto-generated method stub

        Whatsapp whatsappObjFirstTime = Whatsapp.getInstance();
        System.out.println("Memory location of the object :"+
                           whatsappObjFirstTime);

    }

}
```

```
<terminated> Singleton [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (08-May-2024, 5:20:58
Memory location of the object :Java_AdvanceTraining_Day4_Wipro.Whatsapp@5e9f23b4
```

Factory Pattern

```java
package Java_AdvanceTraining_Day4_Wipro;

import java.util.Scanner;

abstract class Plan1{
    double ratePerUnit;
    abstract void getRate();
    public void calculateBill (int units) {
    System.out.println("Consumer has to pay Rs."+units+ratePerUnit);
    }
}

    abstract class DomesticPlan1 extends Plan1{
        public void getRate() {
            ratePerUnit = 3.50;
        }
    }

    class InstitutionPlan1 extends Plan1{
        public void getRate() {
            ratePerUnit = 5.50;
        }
    }

    class CommercialPlan1 extends Plan1{
        public void getRate() {
            ratePerUnit = 7.50;
        }

    class GetPlanFactory1{
        public Plan1 getPlan(String planType) {
            if(planType.equals("Domestic"))
                return new InstitutionPlan1();
             if(planType.equals("Institution"))
                return new InstitutionPlan1();
            else if(planType.equals("Commercial"))
                return new CommercialPlan1();
            return null;
        }
    }

public class FactoryPattern {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Scanner sc = new Scanner(System.in);
        System.out.println
("Enter the type of building : (Domestic/Institution/Commercial)");
            String strBuildingType = sc.next();

        GetPlanFactory1 planFactory = new GetPlanFactory1();
        Plan1 pln = planFactory.getPlan(strBuildingType);

```

```
56
57          System.out.println("How many units of consumed...");
58          int iUnits = sc.nextInt();
59          pln.getRate();
60
61          System.out.println("Rate per Unit for"+ strBuildingType
62                  +"is Rs.:"+pln.ratePerUnit);
63          pln.calculateBill(iUnits);
64
65      }
66
67  }
```

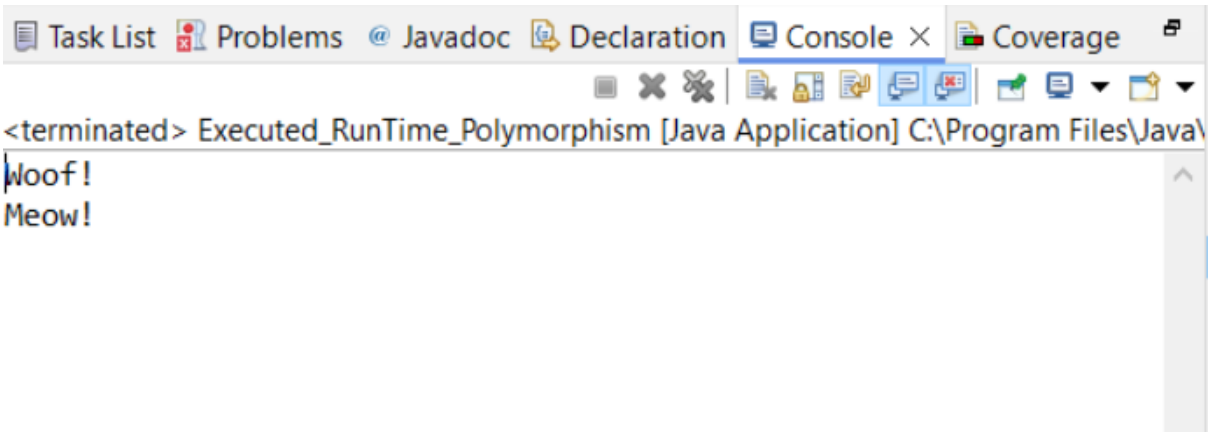
```
<terminated> FactoryPattern [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe
Enter the type of building : (Domestic/Institution/Commercial)

Institution
How many units of consumed...
1
Rate per Unit forInstitutionis Rs.:5.5
Consumer has to pay Rs.15.5
```

Executed RunTime Polymorphism

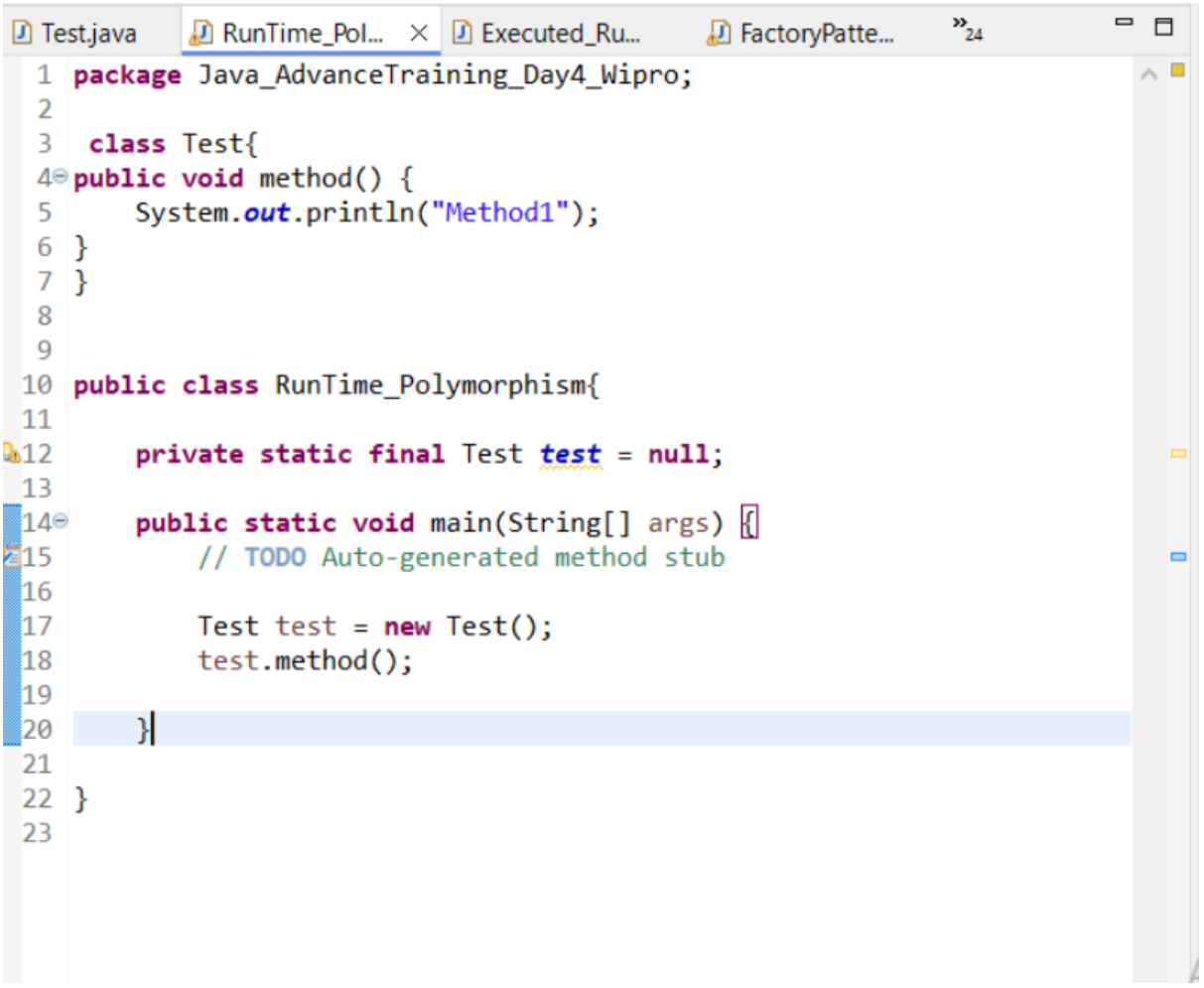
Test.java | RunTime_Pol... | Executed_Ru... | FactoryPatte... | »24

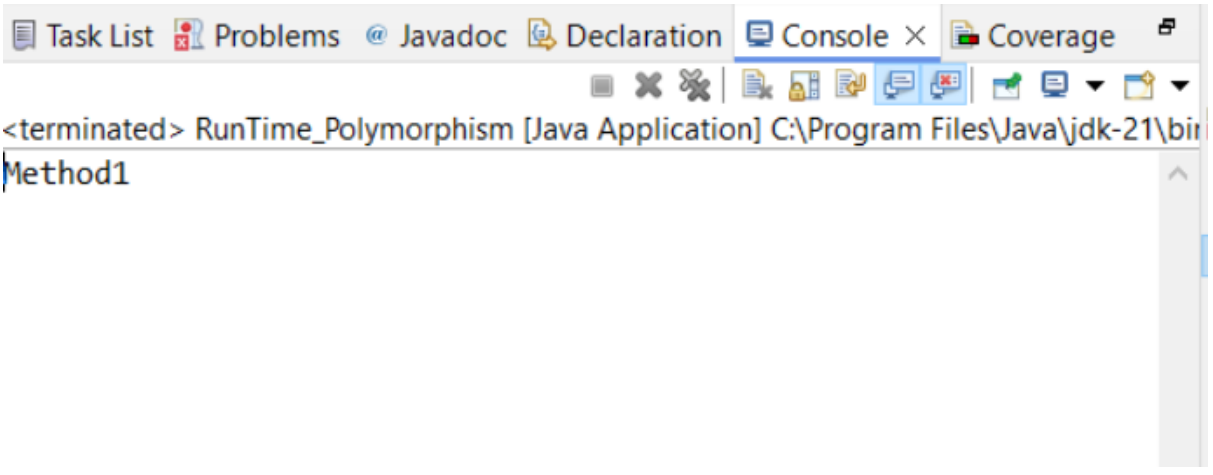
```
1  package Java_AdvanceTraining_Day4_Wipro;
2
3  //Base class Animal
4      class Animal {
5          // Method speak to be overridden by subclasses
6          public void speak() {
7              System.out.println("Animal speaks");
8          }
9      }
10     // Subclass Dog
11     class Dog extends Animal {
12         // Override speak method
13         public void speak() {
14             System.out.println("Woof!");
15         }
16     }
17
18     // Subclass Cat
19     class Cat extends Animal {
20         // Override speak method
21         public void speak() {
22             System.out.println("Meow!");
23         }
24     }
25
26
```

```java

public class Executed_RunTime_Polymorphism {

    public static void makeAnimalSpeak(Animal animal) {
        animal.speak();
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Animal Dog = new Dog();
        Animal Cat = new Cat();

        makeAnimalSpeak(Dog);
        makeAnimalSpeak(Cat);


    }

}

```



RunTime Polymorphism





DAY 5

Builder Pattern

```java
package Java_AdvanceTraining_Day5_Wipro;

class ShoppingItem {
    private String itemName;
    int quantity;
    private double pricePerUnit;

    public ShoppingItem(String itemName, int quantity,
            double pricePerUnit) {
        this.itemName = itemName;
        this.quantity = quantity;
        this.pricePerUnit = pricePerUnit;
    }

    public String getItemName() {
        return itemName;
    }

    public int getQuantity() {
        return quantity;
    }

    public double getPricePerUnit() {
        return pricePerUnit;
    }

    public double getTotalPrice() {
        return quantity * pricePerUnit;
    }

    @Override
    public String toString() {
        return "Item: " + itemName + ", Quantity: " + quantity + ","
                + " Price per unit: $" + pricePerUnit + ", "
                        + "Total Price: $" + getTotalPrice();
    }
}

class ShoppingBuilder {
    private ShoppingItem item;

    public ShoppingBuilder(String itemName, int quantity,
            double pricePerUnit) {
        this.item = new ShoppingItem(itemName, quantity,
                pricePerUnit);
    }

    public ShoppingBuilder setQuantity(int quantity) {
        this.item.quantity = quantity;
        return this;
    }
```

```
    public ShoppingItem build() {
        return this.item;
    }
}

public class BuilderPattern {
    public static void main(String[] args) {
        ShoppingItem groceryItem = new ShoppingBuilder
                ("Grocery", 2, 5.0).build();
        System.out.println(groceryItem);

        ShoppingItem vegetableItem = new ShoppingBuilder
                ("Vegetable", 1, 3.5)
                .setQuantity(3)
                .build();
        System.out.println(vegetableItem);

        ShoppingItem fruitItem = new ShoppingBuilder("Fruit",
                5, 2.0)
                .setQuantity(10)
                .build();
        System.out.println(fruitItem);
    }
}
```



```
<terminated> BuilderPattern [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (08-
Item: Grocery, Quantity: 2, Price per unit: $5.0, Total Price: $10.0
Item: Vegetable, Quantity: 3, Price per unit: $3.5, Total Price: $10.5
Item: Fruit, Quantity: 10, Price per unit: $2.0, Total Price: $20.0
```

Bank Loan

```
package Java_AdvanceTraining_Day5_Wipro;

interface Loan {
    double calculateInterest();
}

class EducationalLoan implements Loan {
    @Override
    public double calculateInterest() {
        return 0.10;
    }
}

class HousingLoan implements Loan {
    @Override
    public double calculateInterest() {
        return 0.12;
    }
}

class BusinessLoan implements Loan {
    @Override
    public double calculateInterest() {
        return 0.15;
    }
}
```

```java
class LoanFactory {
    public static Loan getLoan(String loanType) {
        switch (loanType.toLowerCase()) {
            case "educational":
                return new EducationalLoan();
            case "housing":
                return new HousingLoan();
            case "business":
                return new BusinessLoan();
            default:
                return null;
        }
    }
}

public class Bank_Loan {
    public static void main(String[] args) {
        String loanType = "business"; // You can change this to any type of loan
        Loan loan = LoanFactory.getLoan(loanType);
        if (loan != null) {
            double interestRate = loan.calculateInterest();
            System.out.println("The interest rate for " +
            loanType + " loan is " + (interestRate * 100) + "%");
        } else {
            System.out.println("Invalid loan type");
        }
    }
}
```


```
<terminated> Bank_Loan [Java Application] C:\Program Files\Java\j
The interest rate for business loan is 15.0%
```

DAY 6

Adapter Pattern

```java
package Java_AdvanceTraining_Day6_Wipro;

interface MediaPlayer1{

public void play (String audioType, String fileName); }

interface AdvancedMediaPlayer1{
    public void playVLC (String fileName);
    public void playMp4 (String fileName);
    }

//Step: 2
class VLCPlayer1 implements AdvancedMediaPlayer1{
@Override
public void playVLC (String fileName) {
    System.out.println("Playing VLC file name: "+fileName);

}
@Override
public void playMp4 (String fileName) {
//do nothing.
}
}

class MP4Player1 implements AdvancedMediaPlayer1 {
@Override
public void playVLC (String fileName) {
//do nothing..

}

@Override
public void playMp4 (String fileName) {
System.out.println("Playing MP4 for file name: "+fileName);

}
}

//Step: 3

class MediaAdapter1 implements MediaPlayer1{
AdvancedMediaPlayer1 advanedMediaPlayer;
public MediaAdapter1 (String audioType) {
if (audioType.equalsIgnoreCase ("vlc"))
    advanedMediaPlayer = new VLCPlayer1();

else if (audioType.equalsIgnoreCase("mp4"))
    advanedMediaPlayer = new MP4Player1();

}

@Override
public void play (String audioType, String fileName) {
if (audioType.equalsIgnoreCase ("mp4"))
advanedMediaPlayer.playMp4 (fileName);
else if (audioType.equalsIgnoreCase("vlc"))
```

```
    advanedMediaPlayer.playVLC(fileName);

}
}

//Step: 4
class AudioPlayer1 implements MediaPlayer1 {
   MediaAdapter1 mediaAdapter;
@Override
public void play (String audioType, String fileName) {
if (audioType.equalsIgnoreCase("mp3"))
System.out.println("Playing Mp3 for the file name "+fileName);
else if (audioType.equalsIgnoreCase("vlc") || audioType.equalsIgnoreCase("mp4")) {
    mediaAdapter = new MediaAdapter1 (audioType); //Runtime polymorphism is created..
    mediaAdapter.play(audioType, fileName);
    } //Corresponding file is played..

else
System.out.println("Invalid Media...");
}
}
```

```
public class Adapter_Pattern {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        AudioPlayer1 audioPlayer = new AudioPlayer1();
        audioPlayer.play("mp3", "123 file..");
        audioPlayer.play("mp4", "23456 file..");
        audioPlayer.play("vlc", "vlc file..");
        audioPlayer.play("avi", "avi file");

    }

}
```


```
<terminated> Adapter_Pattern (1) [Java Application] C:\Program Files\Java\jdk-21\
Playing Mp3 for the file name 123 file..
Playing MP4 for file name: 23456 file..
Playing VLC file name: vlc file..
Invalid Media...
```

JSP and Servlet

Scriptlet

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h1>I am Soumyajit. I am from Burdwan, West Bengal </h1>
    <%
    int a = 220, b = 380, c;
    c = a + b;
    out.println("Result is :"+c);
    %>

</body>
</html>
```

I am Soumyajit. I am from Burdwan, West Bengal

Declarative Tags

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1> Factorial using Declarative and Scriptlet tags </h1>

<%!
int factorial(int n){
    int fact =1;
    for(int i =1; i <=n; ++i)
        fact = fact * i;
    return fact;
}
%>
<%
int result = factorial(5);
out.println("Factorial of 5 is : "+ result);
%>

</body>
</html>
```


Factorial using Declarative and Scriptlet tags

Factorial of 5 is : 120

Prime or Not

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Prime or not</title>
</head>
<body>
<h1>Prime Number Checker</h1>
    <form method="post">
        Enter a number: <input type="text" name="number">
        <input type="submit" value="Check">
    </form>

    <%
        String numberStr = request.getParameter("number");
        if (numberStr != null && !numberStr.isEmpty()) {
            int number = Integer.parseInt(numberStr);
            boolean isPrime = true;

```

```
            if (number <= 1) {
                isPrime = false;
            } else {
                for (int i = 2; i <= Math.sqrt(number); i++) {
                    if (number % i == 0) {
                        isPrime = false;
                        break;
                    }
                }
            }
    %>

    <p>The number <%= number %> is <%= isPrime ? "prime" : "not prime" %>.</p>

    <% } %>

</body>
</html>
```


Prime Number Checker

Enter a number: Check

The number 6 is not prime.

Prime Number Checker

Enter a number: Check

The number 2 is prime.

Spring Core

Displaying the bean info

Pom.xml

```
  https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
  <modelVersion>4.0.0</modelVersion>
  <groupId>EricsonTest</groupId>
  <artifactId>EricsonTest</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>EricsonTest</name>
  <description>EricsonTest</description>
  <dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>5.2.16.RELEASE</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.16.RELEASE</version>
</dependency>
  </dependencies>
</project>
```

Employee1.java

```
public class Employee1 {

    int id;
    String name;
    String dept;
    String designation;

    Employee1(){}
        public Employee1(int id, String name, String dept, String designation) {

            this.id = id;
            this.name = name;
            this.dept = dept;
            this.designation = designation;
        }


    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
```

```
28      public String getName() {
29          return name;
30      }
31
32      public void setName(String name) {
33          this.name = name;
34      }
35
36      public String getDept() {
37          return dept;
38      }
39
40      public void setDept(String dept) {
41          this.dept = dept;
42      }
43
44      public String getDesignation() {
45          return designation;
46      }
47
48      public void setDesignation(String designation) {
49          this.designation = designation;
50      }
51
52  }
```


display_Bean_Info.java

```
1  import org.springframework.context.ApplicationContext;
3
4
5  public class display_Bean_Info {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9      ApplicationContext context = new ClassPathXmlApplicationContext("display_BeanInfo_Hellobean.xml");
10
11                 Employee1 empObj = (Employee1) context.getBean("emp");
12
13                 System.out.println("Employee details...");
14
15                 System.out.println("Id: "+empObj.getId());
16
17                 System.out.println("Name: "+empObj.getName());
18                 System.out.println("Dept: "+empObj.getDept());
19
20                 System.out.println("Designation: "+empObj.getDesignation());
21
22                 }
23     }
24
25
```

```
<terminated> display_Bean_Info [Java Application] C:\Program Files\Java\jdk-21\bi
Employee details...
Id: 102
Name: Soumyajit Mondal
Dept: Technology
Designation: Full Stack Developer
```

Spring MVC

My Controller

MyController.java

```java
package com.soumya.demo.controller;

import org.springframework.stereotype.Controller;

@Controller
public class MyController {

    // when the user gives in url as http://localhost:8089/
    // inside the controller , it will be looked for "/" and the function
    // which below it is executed automatically
    @GetMapping ("/")
    public String displayGreeting() {
        return "welcome";
    }

    //http://localhost:8089/login
    @GetMapping("/login")
    public String displayLogin() {
        return "login";   //WEB-INF\jsp\login.jsp
    }

    @PostMapping ("/login")
    public String loginValidation(ModelMap model , @RequestParam String userID,
            @RequestParam String password) {
    System.out.println("Received data : User ID:"+ userID + "password :" + password);
    String str = "Invalid User";

    if(userID.equals("Soumya")&& password.equals("Admin"))
        str = "Valid User";

    model.put("message", str);
    return "results";
}
}
```

Application.properties

```
spring.application.name=SpringMVC
server.port=8089
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp
```

Welcome.jsp File

```
<h1> Hii Sir , Myself Soumyajit. </h1>
```

Hii Sir , Myself Soumyajit.

Login Validation

```
package com.soumya.demo.controller;

import org.springframework.stereotype.Controller;

@Controller
public class MyController {

    // when the user gives in url as http://localhost:8089/
    // inside the controller , it will be looked for "/" and the function
    // which below it is executed automatically
    @GetMapping ("/")
    public String displayGreeting() {
        return "welcome";
    }

    //http://localhost:8089/login
    @GetMapping("/login")
    public String displayLogin() {
        return "login";   //WEB-INF\jsp\login.jsp
    }

    @PostMapping ("/login")
    public String loginValidation(ModelMap model , @RequestParam String userID,
            @RequestParam String password) {
    System.out.println("Received data : User ID:"+ userID + "password :" + password);
    String str = "Invalid User";

    if(userID.equals("Soumya")&& password.equals("Admin"))
        str = "Valid User";

    model.put("message", str);
    return "results";
}
}
```

Login.jsp File

```
<html>
<body>

<h1> Login Form </h1>
<FORM method = "post">
User ID : <Input type = "text" name ="userID"/> <br/> <br/>
Password : <Input type = "text" name ="password"/> <br/> <br/>
<Input type = "Submit"/>

</FORM>
</body>
</html>
```

Results.jsp File

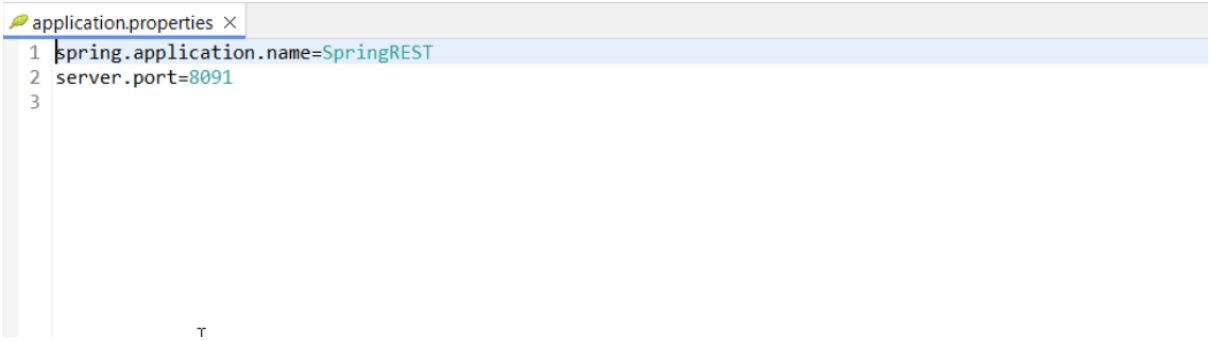
```
<h1>${message}</h1>
```

Run the program in the browser

http://localhost:8089/login

Spring REST

Application.properties



Welcome

Login Validation

```java
package com.soumya.demo.controller;

import org.springframework.web.bind.annotation.DeleteMapping;

@RestController
public class MyController {

    @GetMapping ("/welcome")
    public String welcomeMessage() {
        String str = "Welcome to Spring REST from Soumya...";
        return str;
    }

    //Read the parameter from the client.
    @GetMapping ("/login")
    public String loginValidation(@RequestParam String UID, @RequestParam String pwd) {
        System.out.println("Given credentials are : Login" + UID+ "Password:" +pwd);
        if (UID.equals("Soumya")&& pwd.equals("Admin"))
            return "You are a valid user";
        else
            return "Invalid user !! Check your credentials";
    }
```

Output

You are a valid user

Invalid user !! Check your credentials

POJO Class - Employee.java

```java
package com.soumya.demo.data;

public class Employee {

    int id;
    String name;
    String dept;
    String designation;


    public Employee(int id, String name, String dept, String designation) {
        super();
        this.id = id;
        this.name = name;
        this.dept = dept;
        this.designation = designation;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDept() {
        return dept;
    }

    public void setDept(String dept) {
        this.dept = dept;
    }

    public String getDesignation() {
        return designation;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }


    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ", dept=" + dept + ", designation=" + designation + "]";
    }


}
```

MyController.java

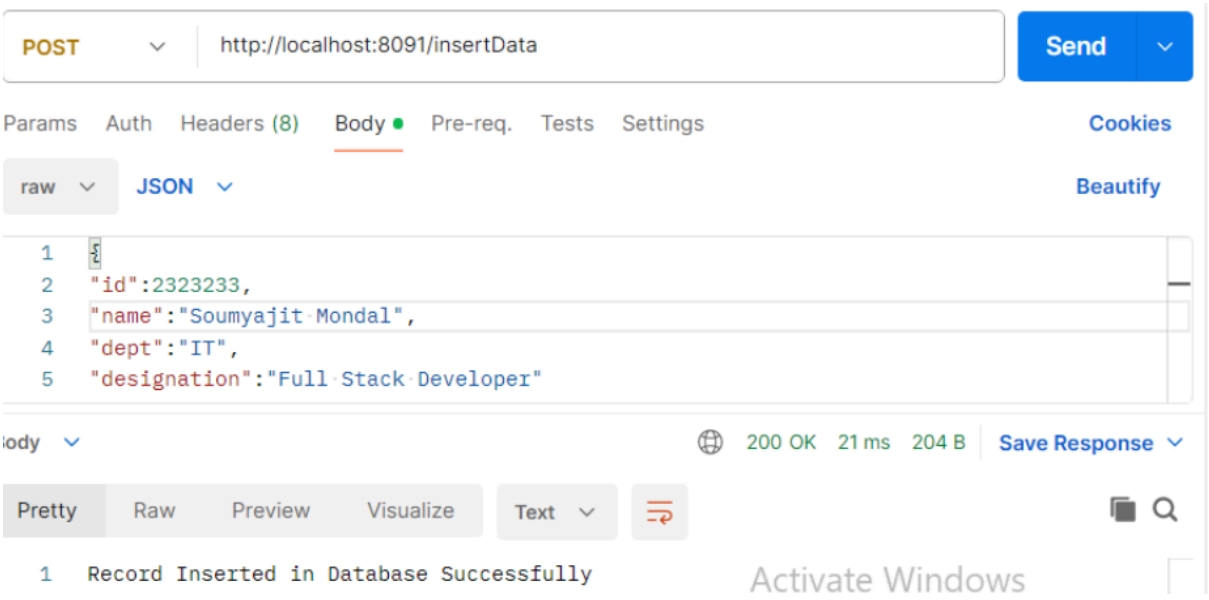
CRUD Operations

```
33
34    @PostMapping("/insertData")
35    public String insertData(@RequestBody Employee empObj) {
36        System.out.println("Given data from client is : " + empObj);
37        return "Given data is inserted in the database successfully...";
38    }
39
40    @PutMapping("/updateData")
41    public String updateData(@RequestBody Employee empObj) {
42        System.out.println("Given Object to update is : "+empObj);
43        return "Given data is updated in the database successfully...";
44    }
45
46    @DeleteMapping("/deleteById/{id}")
47    public String deleteByID(@PathVariable int id) {
48        String strReturn = "Employee with id "+"is deleted successfully from the database";
49        return strReturn;
50    }
51
52 }
53
```

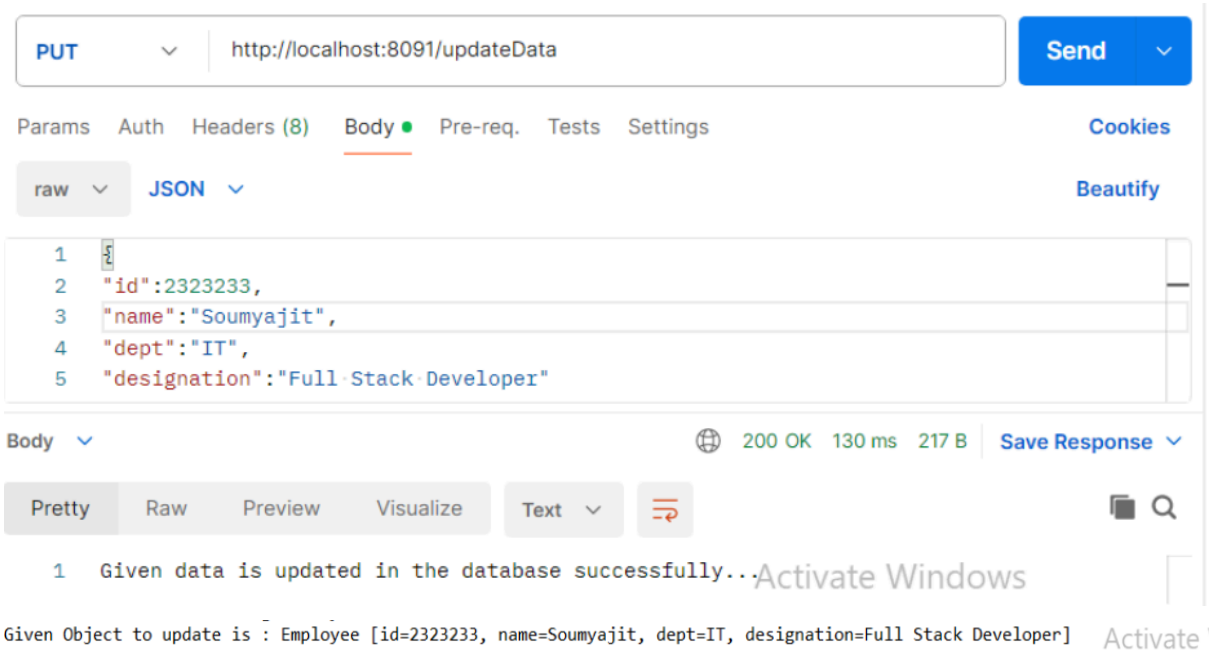

Run the Server

Check POSTMAN

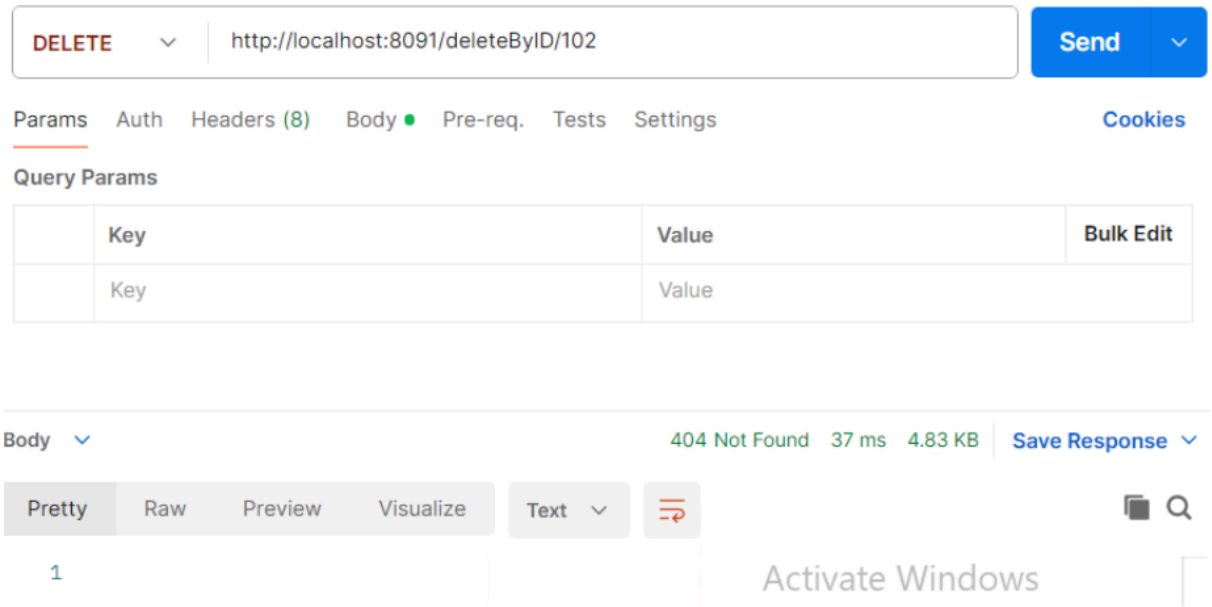
POST Operation



PUT Operation



DELETE Operation



Spring UNIT Test

MyController

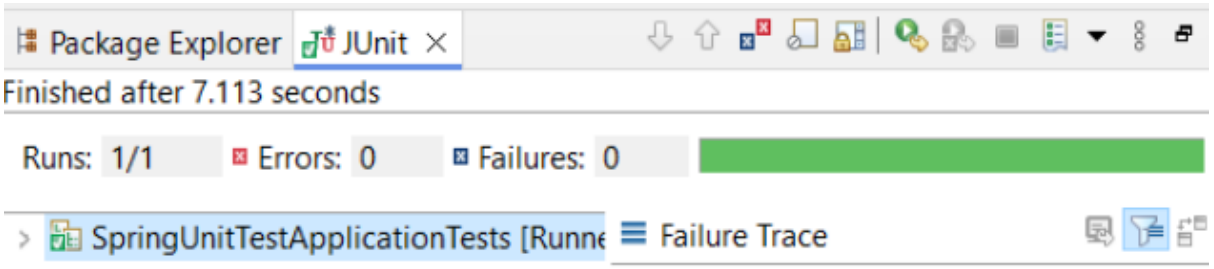
```java
package com.soumya.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;

@RestController

public class MyController {

    @GetMapping("/login")
    public String login(String UID, String PWD) {
        if(UID.equals("Soumya")&& PWD.equals("Admin"))
            return "You are a Valid User";
        else
            return "You are a Invalid User";
    }

}

```

SpringUnitIntegratonTestingTests.java.

SpringUnitTestApplicationTests.java

```java
package com.soumya.demo;

import static org.junit.jupiter.api.Assertions.assertEquals;

@SpringBootTest
class SpringUnitTestApplicationTests {

    @Autowired
    MyController myController;

    @Test
    void contextLoads() {

        String UID="Soumya", PWD="Admin";
        String strValidUser = "You are a Valid User";
        String strInvalidUser= "You are an Invalid User";

        String strResult = myController.login(UID, PWD);
        assertEquals(strValidUser,strResult);
    }

}

```

Output



Hibernate

CRUD Operation

hibernate.cfg.xml File

```
-//Hibernate/Hibernate Configuration DTD 3.0//EN (doctype)
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE hibernate-configuration PUBLIC
3
4  "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
5  "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
6  <hibernate-configuration>
7  <session-factory>
8  <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
9  <property name="hibernate.connection.url">jdbc:mysql://127.0.0.1:3306/test</property>
10 <property name="hibernate.connection.username">root</property>
11  <property name="connection.password"> admin</property>
12 <property name="connection.pool_size">10</property>
13 <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
14 <property name="show_sql">true</property>
15 <property name="hibernate.hbm2ddl.auto">update</property>
16 <mapping resource="employee.hbm.xml"/>
17 </session-factory>
18 </hibernate-configuration>
```

POJO Class

Employee.java

```java
package HibernateCRUD;
public class Employee {

    int id;
    String firstName;
    String lastName;
    int salary;

    Employee(){}

    public Employee(int id, String firstName, String lastName, int salary) {
        super();
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.salary = salary;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }

}
```

employee.hbm.xml File

```
-//Hibernate/Hibernate Mapping DTD//EN (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="Employee" table="hibernate_create_insert_update_delete">
<id name="id" type="int" column="id"> </id>
<property name="firstName" column="first_name" type="string"/>
<                                          "last_name" type="string"/>
<                                          alary" type="int"/>

</class>

</hibernate-mapping>
```

Element type "property" must be declared.
Press 'F2' for focus

Client.java

```
import java.util.List;

import org.hibernate.SQLQuery;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class Client {

    private static SessionFactory factory;


    //Connects with the db.  Creates sessionFactory object.  With this object
    // you can create a session, i.e., a connection is established between client and server.
    public static void getSessionFactory() {

        try {
        Configuration conf = new Configuration().configure();

        StandardServiceRegistryBuilder builder = new StandardServiceRegistryBuilder().applySettings(conf.getProperties());

        factory = conf.buildSessionFactory(builder.build());
        }catch (Exception ex) {
            System.out.println(ex);
        }
    }
```

```
32      public static void main (String args[]) {
33          getSessionFactory();
34
35      Client clientObj = new Client();
36
37      //Inserts record in the db.
38      //clientObj.insertRecordInDatabase (75, "Soumyajit", "Mondal", 40000);
39
40      //Display all the records using HQL klientObj.displayRecords();
41
42      //Display the records using native Query.
43      //clientObj.displayRecordsUsing NativeQuery();
44
45      //Update the data in the db.
46      //clientObj.updateRecord(75, "Soumyajit");
47
48      //Delete record from the db.
49      //clientObj.deleteRecord (13);
50
51      }
52
53      //Inserts a record in the db.
54
```

```
//Display all the records using HQL (Hibernate Query Language)
public void displayRecords() {

    Session session = factory.openSession();


    //HQL - Hibernate Query Language..
    //Select * from hibernate_create_insert_update_delete
    List<Employee> empLst = session.createQuery("FROM Employee").list();

    for (Employee empObj : empLst) {
        System.out.println(empObj);
    }

    session.close();

}

//Displays the records using native SQL QUery.
public void displayRecordsUsingNativeQuery() {
    Session session = factory.openSession();

    String str = "SELECT * from hibernate_create_insert_update_delete WHERE salary >= 10000";
    SQLQuery query = session.createSQLQuery(str);
    query.addEntity(Employee.class);

    List<Employee> empLst = query.list();

    for (Employee empObj : empLst)
        System.out.println(empObj);

    session.close();
}
```

```java
//Update the record in the db.
public void updateRecord (int id, String firstName) {
    Session session = factory.openSession();
    Transaction tx = session.beginTransaction();


    //SELECT * FROM hibernate_create_insert_update_delete WHERE id = given id
    Employee empObj = (Employee) session.get(Employee.class, id);

    empObj.setFirstName(firstName);

    session.update(empObj);

    tx.commit();
    System.out.println("Gien record is upated in the db successfully...");
    session.close();
}

//Deletes the record from the db.
public void deleteRecord (int id) {
    Session session = factory.openSession();
    Transaction tx = session.beginTransaction();

    Employee empObj = (Employee) session.get(Employee.class, id);
    session.delete(empObj);

    tx.commit();
    System.out.println("Record with id "+ id +" is deleted successfully");
    session.close();
}
```

Spring JPA

application.properties

```
spring.application.name=SpringBootSpringJPA
server.port=8091
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/test
spring.datasource.username=root
spring.datasource.password=admin
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
```

Employee.java

```java
package com.soumya.demo.data;

import jakarta.persistence.Entity;

@Entity
@Table(name="Employee")

public class Employee {

    @Id
    int id;

    String name;
    String dept;
    String designation;

    Employee(){}

    public Employee(int id, String name, String dept, String designation) {
        super();
        this.id = id;
        this.name = name;
        this.dept = dept;
        this.designation = designation;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDept() {
        return dept;
    }

    public void setDept(String dept) {
        this.dept = dept;
    }

    public String getDesignation() {
        return designation;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }


}
```

Employee Repository

EmployeeRepository.java

```java
package com.soumya.demo.repository;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends JpaRepository <Employee,  Integer> {



}

```

Employee Controller

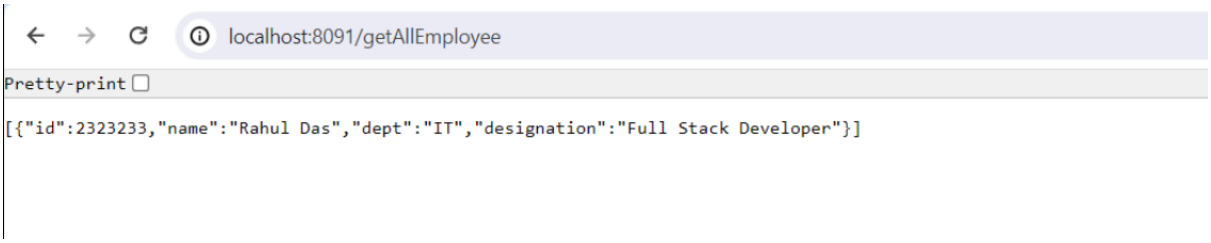
EmployeeController.java

```java
package com.soumya.demo.controller;

import java.util.List;

@RestController
public class EmployeeController {

    @Autowired
    EmployeeRepository empRepository;

    @GetMapping("getAllEmployee")
    public List<Employee> getAllEmployees(){
        List<Employee> empLst = empRepository.findAll();
        return empLst;
    }

    //Insert the data
    @PostMapping("/insertData")
    public String insertData(@RequestBody Employee empObj) {
        System.out.println("Given Object is : "+empObj);
        empRepository.save(empObj);

        String strReturn = "Record Inserted in Database Successfully";
        return strReturn;
    }

```

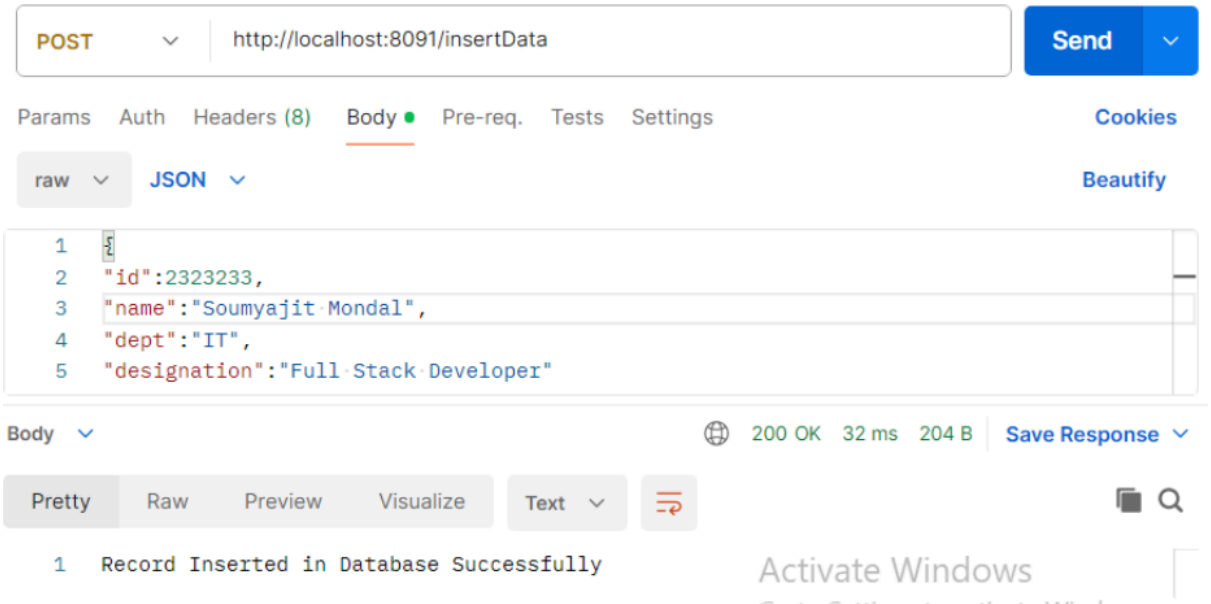
```java
    @PutMapping ("/updateData")
    public String updateData(@RequestBody Employee empObj) {
        System.out.println("Given Object to Update is : "+empObj);
        empRepository.save(empObj);

        String strReturn = "Record Updated in Database Successfully";
        return strReturn;
    }

    @DeleteMapping("/deleteData/{id}")
    public String deleteData(@PathVariable Employee empObj) {
        System.out.println("Given Object to Delete is : "+empObj);
        empRepository.save(empObj);

        String strReturn = "Record Deleted in Database Successfully";
        return strReturn;
    }

}
```


CRUD Operation

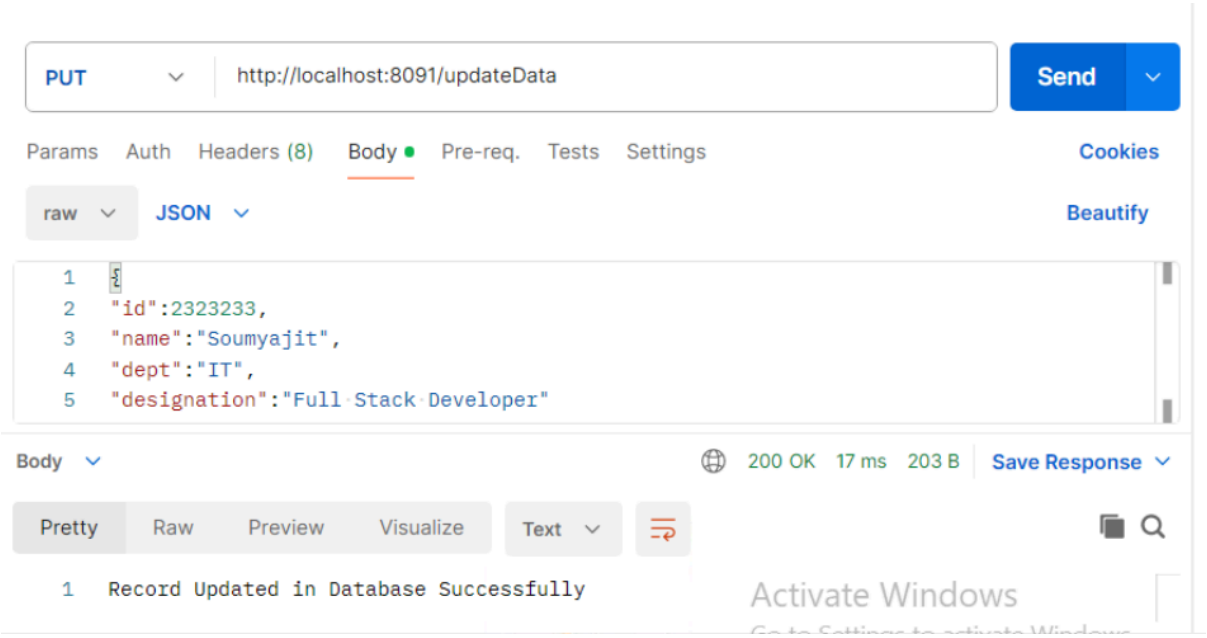
GET Data



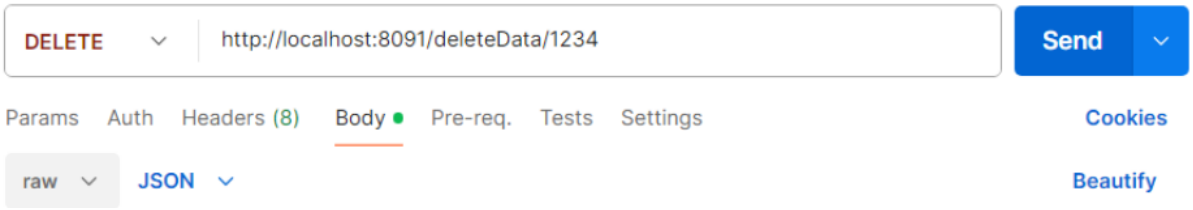
POST _ Insert Data



PUT Data



Delete Data



Spring MicroServices

Eureka Server Enable

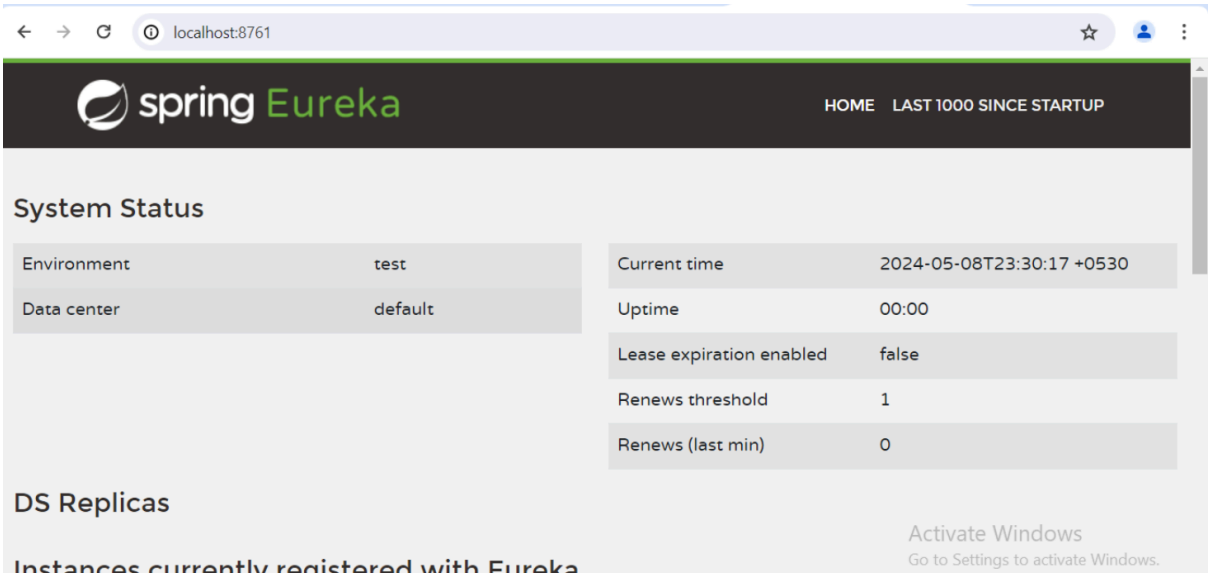
SpringEurekaServerApplication.java

```java
package com.soumya.demo;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@EnableEurekaServer
public class SpringEurekaServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringEurekaServerApplication.class, args);
    }

}
```

application.properties

```
spring.application.name=SpringEurekaServer
server.port=8761
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
```

Server is running or not



Eureka Client Employee

application.properties

```
spring.application.name=SpringEurekaClientEmployeeData1
server.port=8082
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
eureka.client.healthcheck.enabled=true
eureka.client.fetch-registry=true
```

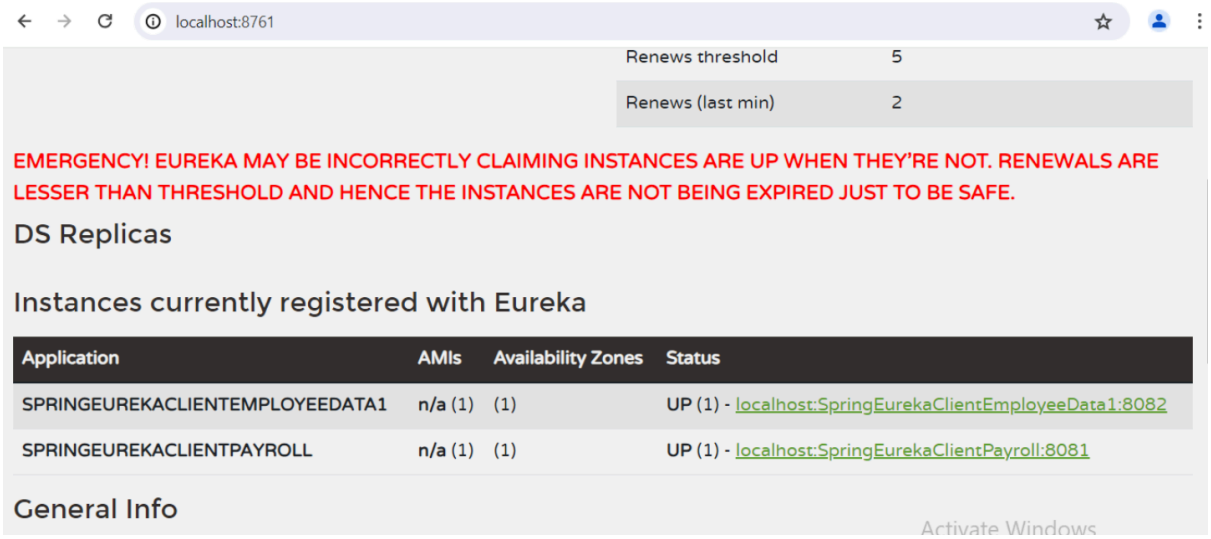
Enable Discovery Client

```java
package com.soumya.demo;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@EnableDiscoveryClient
public class SpringEurekaClientEmployeeData1Application {

    public static void main(String[] args) {
        SpringApplication.run(SpringEurekaClientEmployeeData1Application.class, args);
    }

}
```

MyController

```java
package com.soumya.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;

@RestController
public class MyController {

    @GetMapping("/getDataWithID/{id}")
    public String getDataWithID(@PathVariable int id) {
        System.out.println("Received ID from Client is : "+ id);
        if(id == 101)
            return "Soumya";
        else if(id == 102)
            return "Charan Sir";
        else if (id == 103)
            return "Sakshi";
        else if (id == 104)
            return "Ankita";
        else if (id == 105)
            return "Trupti";

        return "Invalid Employee";

    }

}
```

Run the program



Client PayRoll

Application.properties

```
spring.application.name=SpringEurekaClientPayroll
server.port=8081
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
eureka.client.healthcheck.enabled=true
eureka.client.fetch-registry=true

```

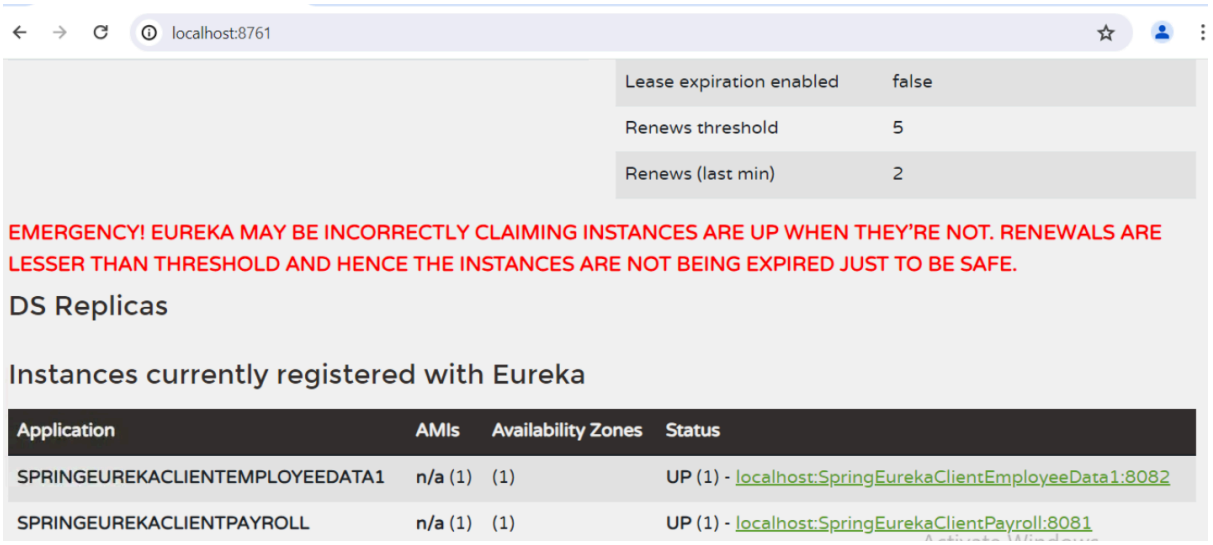
ClientPayrollApplication.java

```
package com.soumya.demo;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@EnableDiscoveryClient
public class SpringEurekaClientPayrollApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringEurekaClientPayrollApplication.class, args);
    }

    @Bean
    @LoadBalanced
    public RestTemplate restTemplate(RestTemplateBuilder builder) {
        return builder.build();
    }

}

```

MyController

```java
package com.soumya.demo.controller;

import org.springframework.beans.factory.annotation.Autowired;

@RestController
public class EmployeeController {

    @Autowired
    RestTemplate restTemplate;

    @GetMapping ("/getEmployeeData/{id}")
    public String getEmployeeData(@PathVariable int id) {
        System.out.println("Obtained ID from the User is : " + id);

        String strURL = "http://EMPLOYEE/getDataWithID/"+ id;
        String str = restTemplate.getForObject(strURL, String.class);
        return str;
    }
```

Client-Registered



Spring Reactor

testReactor

testReactor.java

```java
package testReactor;

import reactor.core.publisher.Mono;

public class testReactor {

    public Mono<String> nameMono(){
        return Mono.just("Soumyajit Mondal").log();
    }

    public static void main(String[] args) {
        testReactor testObj = new testReactor();
        testObj.nameMono().subscribe(name -> System.out.println("My name is " + name));
    }

}
```

Output

```
<terminated> testReactor [Java Application] C:\software\sts-4.22.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_
[ INFO] (main) | onSubscribe([Synchronous Fuseable] Operators.ScalarSubscription)
[ INFO] (main) | request(unbounded)
[ INFO] (main) | onNext(Soumyajit Mondal)
My name is Soumyajit Mondal
[ INFO] (main) | onComplete()
```

SpringReactorMono

SpringReactorMonoApplication.java

```java
package com.soumya.demo;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SpringReactorMonoApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringReactorMonoApplication.class, args);
    }

}
```

MyController

```java
package com.soumya.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import reactor.core.publisher.Mono;

@RestController
public class MyController {

    @GetMapping("getName")
    public Mono<String> getName(){
        return Mono.just("My name is Soumyajit Mondal").log();

    }

}
```

Output