Advanced Java

Subject Code: MCAL12

A Practical Journal Submitted in Fulfilment of the Degree of

MASTER

In

COMPUTER APPLICATION

Year 2023-2024

By

Mr. KAMBLE TEJAS GUNAJI ANITA

(Application Id: - 74610)

Under the Guidance of
Prof. Mr.DYANESHWAR DEORE

Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

University of Mumbai

PCP Center

[Satish Pradhan Dyanasadhana College, Thane]

Institute of Distance and Open Learning

Vidyanagari, Kalina, Santacruz (E) -400098

CERTIFICATE

This to certify that, **Mr. KAMBLE TEJAS ANITA** appearing **Master in Computer Application (Semester I) Application ID: 74610** has satisfactorily completed the prescribed Practical of **MCAL12- Advanced Java** as laid down by the University of Mumbai for the academic year **2023- 24**

Teacher in charge   Examiners   Coordinator

                                                            IDOL, MCA

                                                            University of Mumbai

Date: -

Place**: -**

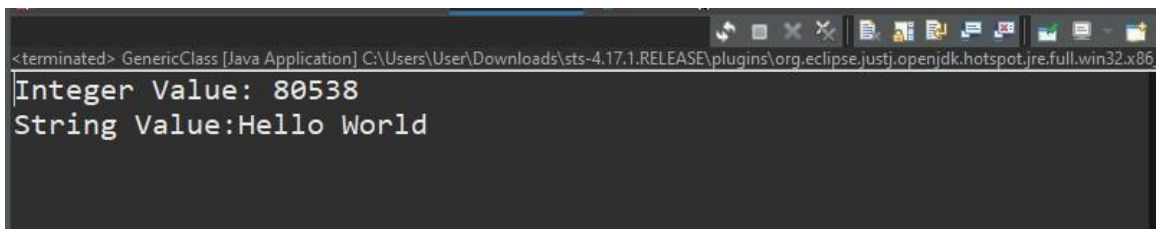| Practical No	Practical
1	**Java Generics** 1. Write a Java Program to Demonstrate a Generic Class 2. Write a Java Program to demonstrate Wildcards in Java Generics.
2	**List Interface** Write a Java program to create List containing list of items of type String and use for---each loop to print the items of the list
3	**Set Interface** Write a Java program using Set interface containing list of items and performthe following operations: a. Add items in the set. b. Insert items of one set in to other set. c. Remove items from the set d. Search the specified item in the set
4	**Lambda Expression** Write a Java program using Lambda Expression to print "Hello World".
5	**JSP** a. Write a JSP program that demonstratesthe use of JSP declaration, scriptlet, directives, expression, header and footer b. Write a JSP page to display the Registration form

Practical No. 1

Aim: - Write a Java Program to Demonstrate a Generic Class

Code: -

```java
package task;
public class GenericClass <T>{
private T t;
public void add(T t){
this.t=t;
}
public T get(){
return t;
}
public static void main(String args[]){
GenericClass<Integer> intObj = new GenericClass<Integer>();
GenericClass<String> strObj = new GenericClass<String>();
intObj.add(new Integer(80538));
strObj.add(new String("Hello World"));
System.out.println("Integer Value: "+ intObj.get());
System.out.println("String Value:" + strObj.get());
}
}
```

Output:


```
<terminated> GenericClass [Java Application] C:\Users\User\Downloads\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
Integer Value: 80538
String Value:Hello World
```

Practical No.2

Aim: - Write a Java Program to demonstrate Wildcards in Java Generics.

Code: -

```
package task;

import java.util.Arrays;

import java.util.List;

public class WildCard_1 {

public static void main(String args[]){

// Lower bounded Integer List

List<Integer> list1 =Arrays.asList(1,2,3,4);

// Integer List object is being passed

print(list1);

//Lower bounded Number List

List<Number> list2= Arrays.asList(5,6,7,8);

print(list2);

}

public static void print(List<? super Integer>list){

System.out.println("Output is:"+ list);

}

}
```
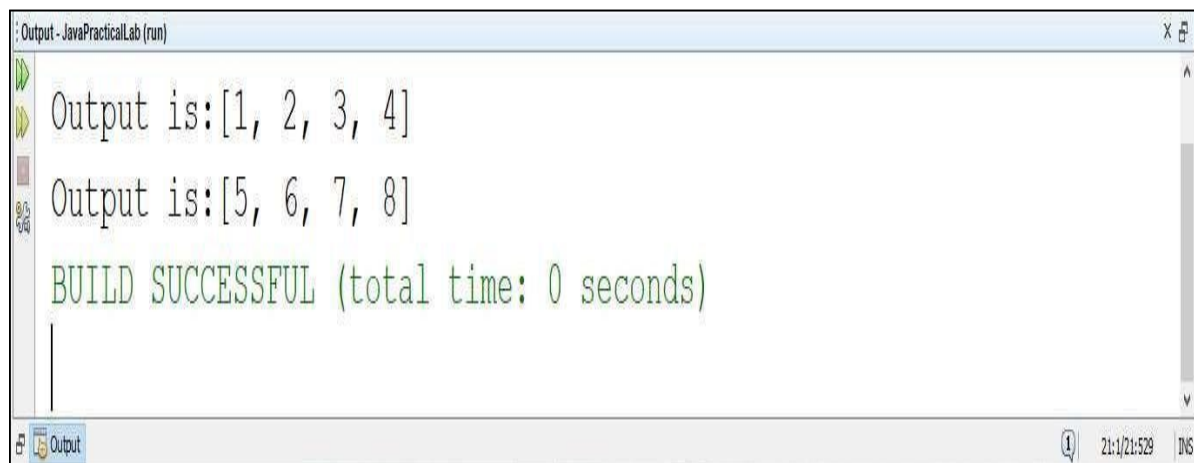
Output:-


```
Output is:[1, 2, 3, 4]

Output is:[5, 6, 7, 8]

BUILD SUCCESSFUL (total time: 0 seconds)
```
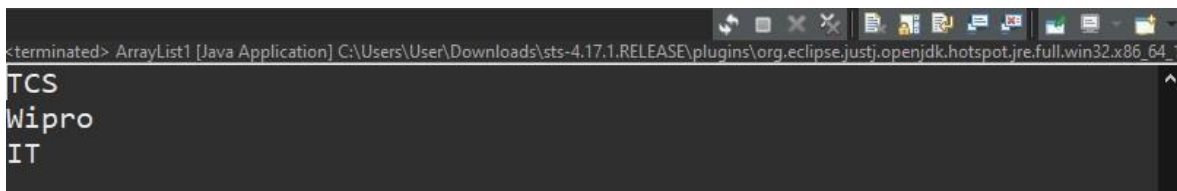
Practical No.3

Assignments on List Interface

Aim: - Write a Java program to create List containing list of items type String and use for each loop to print the items of the list.

Code: -

```
package task;

import java.util.*;

public class ArrayList1 {

public static void main(String args[]){

List<String> str = new ArrayList<String>();

str.add("TCS");

str.add("Wipro");

str.add("IT");

//Using the get method and for loop

for(int i=0;i<str.size();i++){

System.out.println(str.get(i)+"");

}

System.out.println();

//using the for each loop

for(String name: str)

{

System.out.println(name+"");

}

}

}
```

Output: -


```
<terminated> ArrayList1 [Java Application] C:\Users\User\Downloads\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_1
TCS
Wipro
IT
```

Practical No.4

Assignments on Set Interface

Aim: -Write a Java program using Set interface containing list of items and perform the following operations:

    a. Add items in the set.
    b. Insert items of one set in to other.
    c. Remove items from the set.
    d. Search the specified item in the set.

Code: -

```
package MCA_I;

import java.util.*;

public class SetOperations {

public static void main(String args[]){

Integer[] A = {22,45,64,55,87,45,42};

Integer[] B = {34,54,76,75,75,13,35};

Set<Integer> set1 = new HashSet<Integer>();

set1.addAll(Arrays.asList(A));

Set<Integer> set2 = new HashSet<Integer>();

set2.addAll(Arrays.asList(B));


// Finding Union of set1 and set2

Set<Integer> union_data = new HashSet<Integer>(set1);

union_data.addAll(set2);

System.out.print("Union of set1 and set2 is:");

System.out.println(union_data);


// Finding Intersection of set1 and set2

Set<Integer> intersection_data = new HashSet<Integer>(set1);

intersection_data.retainAll(set2);

System.out.print("Intersection of set1 and set2 is:");
```
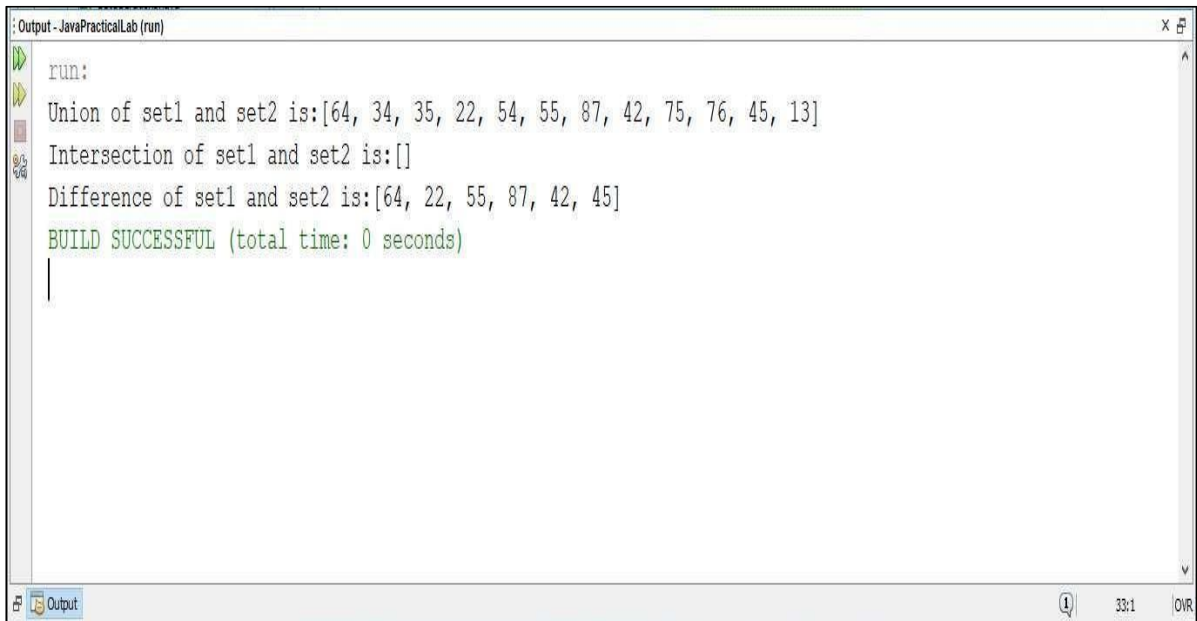
```
System.out.println(intersection_data);

// Finding Diffrence of set1 and set2

Set<Integer> difference_data = new HashSet<Integer>(set1);

difference_data.removeAll(set2);

System.out.print("ifference of set1 and set2 is:");

System.out.println(difference_data);

}

}
```

Output: -

```
Output - JavaPracticalLab (run)                                                    × 🗗
 run:
 Union of set1 and set2 is:[64, 34, 35, 22, 54, 55, 87, 42, 75, 76, 45, 13]
 Intersection of set1 and set2 is:[]
 Difference of set1 and set2 is:[64, 22, 55, 87, 42, 45]
 BUILD SUCCESSFUL (total time: 0 seconds)
```
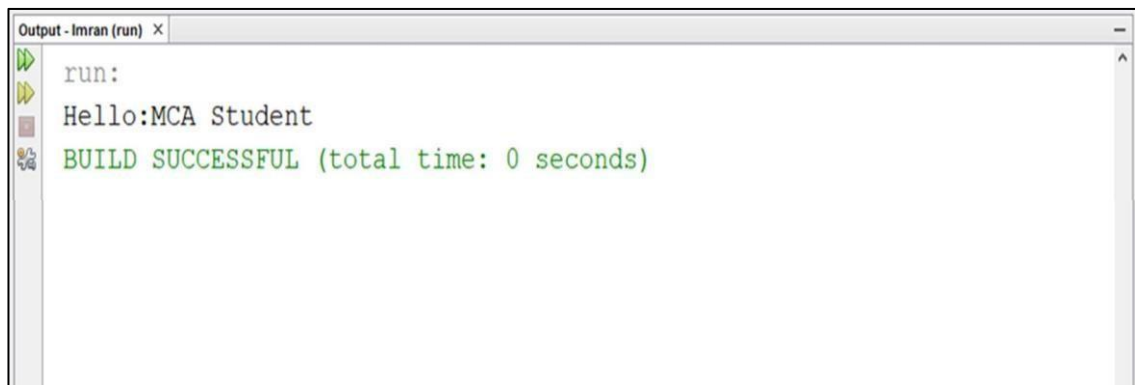
Practical No.5

Aim: - Lambda Expression

Code: -

```
interface abc
{
public void say(String s1, String s2);
}
public class Myexpression {
public static void main(String []args)
{
abc msg=(s1,s2)->
{
System.out.println(s1+""+s2);
};
msg.say("Hello:", "MCA Student");
}
}
```

Output: -

```
Output - Imran (run) ×
run:
Hello:MCA Student
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practical No.6

Aim: - Write a JSP program that demonstratesthe use of JSP declaration, scriptlet, directives, expression, header and footer

Code: -

```
<%--

   Document   : pract06

   Created on : 7 Jan, 2024, 10:41:29 AM

   Author     : spdc

--%>


<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

 <head>

     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

     <title>JSP Page</title>

 </head>

 <body>

     <h1>JSP Scriplet</h1>

     <h1> Declaration Scriplet</h1>

     <%! int a,b,c,m,d,s;%>


      <h1>Simple Scriplet</h1>

     <%

        a=400;

        b=500;

        c=a+b;

        m=a*b;
```
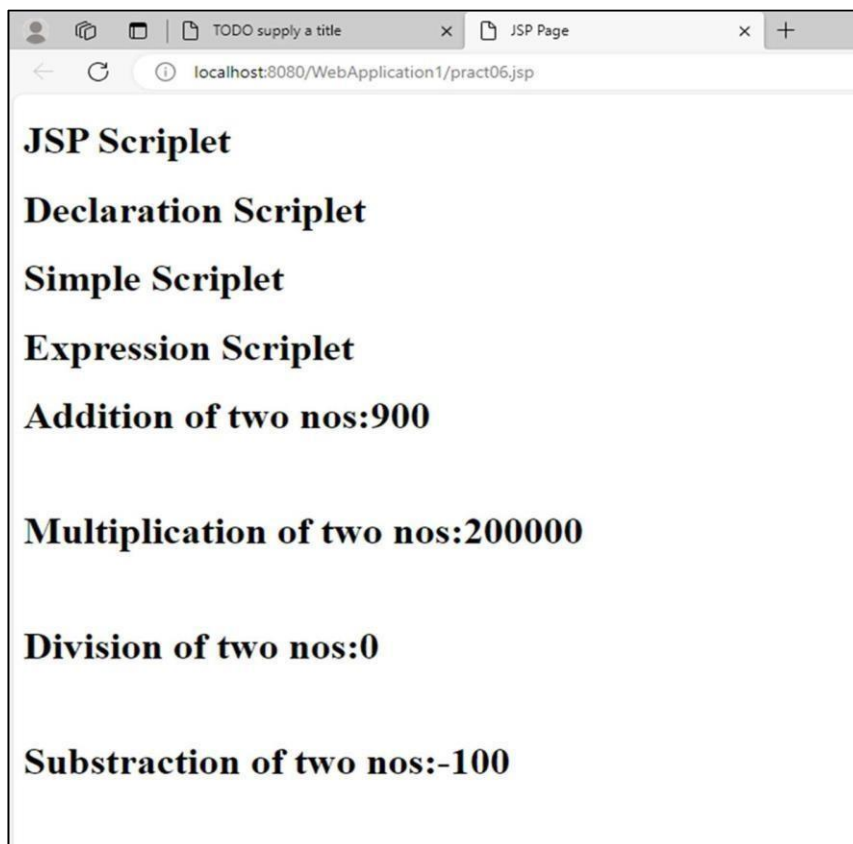
```
        d=a/b;

        s=a-b;

      %>


      <h1>Expression Scriplet</h1>

      <h1><%="Addition of two nos:"+c%></h1><br>

      <h1><%="Multiplication of two nos:"+m%></h1><br>

      <h1><%="Division of two nos:"+d%></h1><br>

      <h1><%="Substraction of two nos:"+s%></h1><br>

   </body>

   </html>
```

Output: -

Practical No.7

Aim: - Write a JSP program to Create a registration form which is stored in database.

Code: -

Index.html

```html
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
    <h1>Registration From</h1>
    <form action="data.jsp">
        <table border="1" width="50" cellspacing="4" cellpadding="4">

          <tbody>
            <tr>
              <td>Username:</td>
              <td><input type="text" name="uname" value="" /></td>

            </tr>
            <tr>

              <td>Password:</td>
              <td><input type="password" name="pass" value="" /></td>

            </tr>
            <tr>
```

```
                <td>Email ID::</td>
                <td><input type="email" name="email" value="" /></td>


        </tr>
        <tr>
          <td>Mobile No:</td>
          <td><input type="number" name="mno" value="" /></td>


        </tr>
        <tr>
          <td><input type="submit" value="REGISTER"/></td>
          <td><input type="reset" value="CLEAR"/></td>
        </tr>


      </tbody>
  </table>
  </form>
  </body>
  </html>
```

Data.jsp

```
<%@page contentType="text/html" import="java.sql.*;"pageEncoding="UTF-
8"%>
<!DOCTYPE html>
<html>


  <body>
```

ADVANCE JAVA

```jsp
        <%
            String uname = request.getParameter("uname");

            String pass = request.getParameter("pass");

            String email = request.getParameter("email");

            String mno = request.getParameter("mno");


            String url = "jdbc:mysql://localhost:3306/db1";


            try {

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

                Connection con = DriverManager.getConnection(url, "root",
    "12345");

                PreparedStatement ps = con.prepareStatement("insert into tblUser
    values(?,?,?,?)");

                ps.setString(1, uname);

                ps.setString(2, pass);

                ps.setString(3, email);

                ps.setString(4, mno);

                ps.executeUpdate();

                out.println("Record Saved Successfully");

            } catch (Exception ex) {

                out.println(ex.getMessage());

            }
        %>
    </body>
    </html>
```

Output: -