



Shiromani Gurdwara Prabandhak Committee's (S.G.P.C.)
GURU NANAK INSTITUTE OF MANAGEMENT STUDIES
(Management Institute of G N Khalsa College),
Matunga, Mumbai – 400 019
Affiliated to University of Mumbai

Name : Khan Tabish Mujeeb

Roll No : 25MCA-33

Subject code : MCAL12

Subject Name : ADVANCE JAVA LAB



Shiromani Gurdwara Prabandhak Committee's (S.G.P.C.)
GURU NANAK INSTITUTE OF MANAGEMENT STUDIES

(Management Institute of G N Khalsa College),
Matunga, Mumbai – 400 019
Affiliated to University of Mumbai

Advanced Java Lab

Index

Practical Number	Problem Statement	Date
1	Assignments on Java Generics 1. Write a Java Program to demonstrate a Generic Class. 2. Write a Java Program to demonstrate Generic Methods. 3. Write a Java Program to demonstrate Wildcards in Java Generics.	23/09/25
2	Assignments on List Interface 1. 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. 2. Write a Java program to create List containing list of items and use List Iterator interface to print items present in the list. Also print the list in reverse / backward direction.	29/09/25
3	Assignments on Set Interface 1. Write a Java program to create a Set containing list of items of type String and print the items in the list using Iterator interface. Also print the list in reverse / backward direction. 2. 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 set. c. Remove items from the set d. Search the specified item in the set	30/09/25
4	Assignments on Map Interface 1. Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations: a. Add items in the map. b. Remove items from the map c. Search specific key from the map d. Get value of the specified key e. Insert map elements of one map in to other map. f. Print all keys and values of the map.	06/10/25
5	Assignments on Lambda Expression	07/10/25

	<p>1. WAP using Lambda Expression to print “Hello World”.</p> <p>2. Write a Java program using Lambda Expression to concatenate two strings.</p> <p>3. WAP using Lambda Expression with single parameters.</p> <p>4. Write a Java program using Lambda Expression with multiple parameters to add two numbers.</p> <p>5. Write a Java program using Lambda Expression to calculate the following:</p> <ul style="list-style-type: none"> a. Convert Fahrenheit to Celsius b. Convert Kilometers to Miles. <p>6. Write a Java program using Lambda Expression with or without return keyword.</p>	
6	Assignments based on web application development using JSP	27/10/25
	<p>1. Create a Telephone directory using JSP and store all the information within a database, so that later could be retrieved as per the requirement. Make your own assumptions.</p> <p>2. Write a JSP page to display the Registration form (Make your own assumptions)</p> <p>3. Write a JSP program to add, delete and display the records from Student Master (RollNo, Name, Semester, Course) table.</p> <p>4. Design Loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount. Display the payment amount for each loan and then list the loan balance and interest paid for each payment over the term of the loan for the following time period and interest rate:</p> <ul style="list-style-type: none"> a) 1 to 7 year at 5.35% b) 8 to 15 year at 5.5% c) 16 to 30 year at 5.75% <p>5. Write a program using JSP that displays a webpage consisting an Application form for change of Study Center which can be filled by any student who wants to change his/ her study center. Make necessary assumptions.</p> <p>6. Write a JSP program that demonstrates the use of JSP declaration, Scriptlet, directives, expression, header and footer.</p>	
7	Assignment based Spring Framework	04/11/25
	<p>1. Write a program to print “Hello World” using spring framework.</p> <p>2. Write a program to demonstrate dependency injection via setter method.</p> <p>3. Write a program to demonstrate dependency injection via Constructor.</p> <p>4. Write a program to demonstrate Autowiring.</p>	
8	Assignment based Aspect Oriented Programming	08/12/25

	<ol style="list-style-type: none"> 1. Write a program to demonstrate Spring AOP – before advice. 2. Write a program to demonstrate Spring AOP – after advice. 3. Write a program to demonstrate Spring AOP – around advice. 4. Write a program to demonstrate Spring AOP – after returning advice. 5. Write a program to demonstrate Spring AOP – after throwing advice. 6. Write a program to demonstrate Spring AOP – pointcuts. 	
9	Assignment based Spring Boot and RESTful Web Services	18/11/25
	<ol style="list-style-type: none"> 1. Write a program to create a simple Spring Boot application that prints a message. 2. Write a program to demonstrate RESTful Web Services with spring boot. 3. Write a program to demonstrate Database Connection with spring boot. 	

PRACTICAL NO. 01**Aim :-****1.1 Write a Java program to demonstrate a Generic Class.****Code :-**

```
package prac1;

public class prac1a<T> {

    private T value;

    public prac1a(T value) {
        this.value = value;
    }

    public void displayValue() {
        if (value instanceof Integer) {
            System.out.println("Value of Integer: " + value);
        } else if (value instanceof String) {
            System.out.println("Value of String: " + value);
        } else {
            System.out.println("Value: " + value);
        }
    }

    public static void main(String[] args) {
        prac1a<Integer> intInstance = new prac1a<>(33);
        intInstance.displayValue();

        prac1a<String> strInstance = new prac1a<>"TABISH KHAN";
        strInstance.displayValue();
    }
}
```

Output :-

The screenshot shows a Java application running in the Eclipse IDE. The code in the main method creates an instance of a class named `prac1a` with an integer value of 33, and then prints the value. The output window displays the results of the execution.

```
16 public static void main(String[] args) {  
17     prac1a<Integer> intInstance = new prac1a<>(33);  
18     intInstance.displayValue();  
19 }  
Console <terminated> prac1a [Java Application] C:\Users\student\.p2\pool\plugins\org.eclipse.jdt.core\org.eclipse.jdt.core_3.20.0.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 9:51:31 am - 9:51:31 am elapsed: 0:00:00.127) [pid: 4604]  
Value of Integer: 33  
Value of String: TABISH KHAN
```

Activate Windows
Go to Settings to activate Windows.

Aim :-

1.2 Write a program to demonstrate Generic Methods.

Code :-

```
package prac1;

public class prac1b {

    void display() {
        System.out.println("GENERIC METHOD EXAMPLE");
    }

    public <T> void printName(T name) {
        System.out.println("Name: " + name);
    }

    public <U> void printRollNo(U rollNo) {
        System.out.println("Roll No: " + rollNo);
    }

    public <V> void printAge(V age) {
        System.out.println("Age: " + age);
    }

    public static void main(String[] args) {
        prac1b obj = new prac1b();
        obj.display();
        obj.printName("TABISH KHAN");
        obj.printRollNo(33);
        obj.printAge(22);
    }
}
```

Output :-

```
--  
29 }  
30 <  
Console X  
<terminated> prac1b [Java Application] C:\Users\student\.p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.v86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 9:58:41 am - 9:58:41 am elapsed: 0:00:00.221) [pid: 17212]  
GENERIC METHOD EXAMPLE  
Name: TABISH KHAN  
Roll No: 33  
Age: 22  
  
Activate Windows  
Go to Settings to activate Windows.
```

Aim :-

1.3 Write a program to demonstrate Wild Cards in Java Generics.

Code :-

```
import java.util.*;

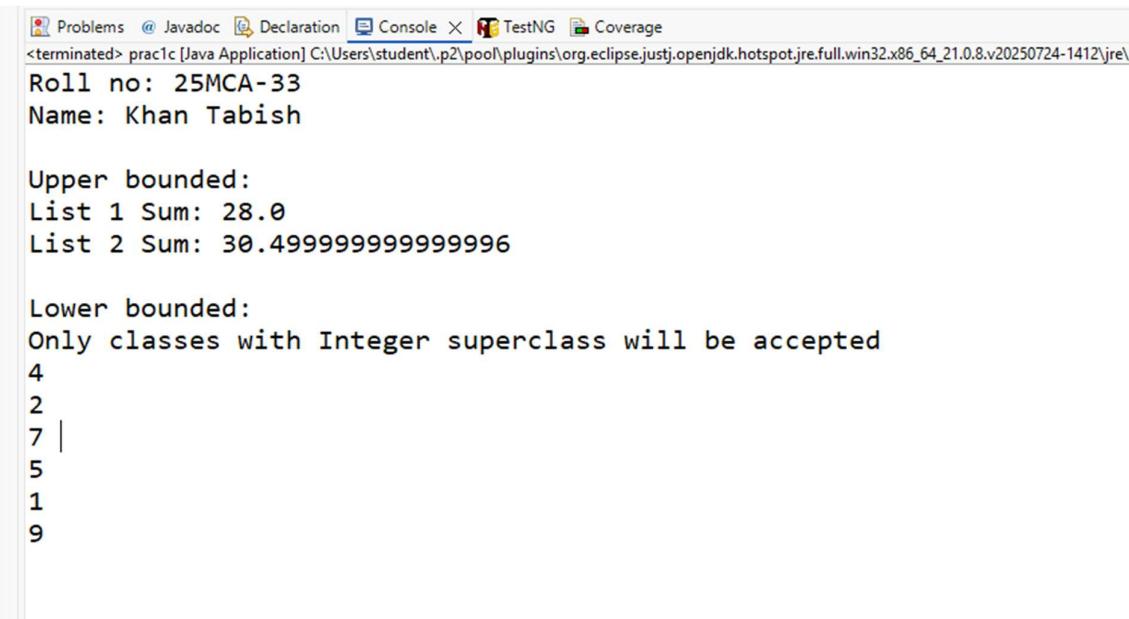
public class prac1c {

    private static double sum(List<? extends Number> list) {
        double sum = 0.0;
        for (Number i : list) {
            sum = sum + i.doubleValue();
        }
        return sum;
    }

    public static void show(List<? super Integer> list) {
        list.forEach(x -> {
            System.out.println(x + " ");
        });
    }

    public static void main(String[] args) {
        System.out.println("Roll no: 25MCA-33");
        System.out.println("Name: Khan Tabish");
        System.out.println("\nUpper bounded:");
        List<Integer> list1 = Arrays.asList(4, 2, 7, 5, 1, 9);
        System.out.println("List 1 Sum: " + sum(list1));
        List<Double> list2 = Arrays.asList(4.7, 2.4, 7.3, 5.4, 1.5, 9.2);
        System.out.println("List 2 Sum: " + sum(list2));
        System.out.println("\nLower bounded:");
        System.out.println("Only classes with Integer superclass will be accepted");
    }
}
```

```
        List<Integer> list3 = Arrays.asList(4, 2, 7, 5, 1, 9);
        show(list3);
    }
}
```

Output :-

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```
Problems Javadoc Declaration Console X TestNG Coverage
<terminated> prac1c [Java Application] C:\Users\student\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\
Roll no: 25MCA-33
Name: Khan Tabish

Upper bounded:
List 1 Sum: 28.0
List 2 Sum: 30.49999999999996

Lower bounded:
Only classes with Integer superclass will be accepted
4
2
7 |
5
1
9
```

PRACTICAL NO. 02**Aim :-**

2.1 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.

Code :-

```
package prac2;  
import java.util.ArrayList;  
  
public class prac2a {  
  
    public static void main(String[] args) {  
  
        ArrayList<String>list = new ArrayList<String>();  
  
        list.add("MATHS");  
  
        list.add("ADBMS");  
  
        list.add("JAVA");  
  
        list.add("PYTHON");  
  
        System.out.println("NAME:TABISH KHAN");  
  
        System.out.println("ROLL-NO:25MCA-33");  
  
        System.out.println(list);  
  
        System.out.println("Traversing list THrough for Each loop");  
  
        for(String subject:list)  
  
            System.out.println(subject);  
  
    }  
}
```

Output :-

The screenshot shows the Eclipse IDE's Console view. The output window title is "Console X". The log message is as follows:

```
16      }
17      }
18      }

<terminated> prac2a [Java Application] C:\Users\student\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 10:04:32 am - 10:04:32 am elapsed: 0:00:00.253) [pid: 14816]
NAME: TABISH KHAN
ROLL-NO:25MCA-33
[MATHS, ADBMS, JAVA, PYTHON]
Traversing list THrough for Each loop
MATHS
ADBMS
JAVA
PYTHON
```

At the bottom right of the console window, there is a watermark that reads "Activate Windows Go to Settings to activate Windows."

Aim :-

2.2 Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse / backward direction.

Code :-

```
package prac2;  
import java.util.*;  
  
public class prac2b {  
  
    public static void main(String[] args) {  
  
        System.out.println("ROLLNO:25MCA-33");  
  
        System.out.println("NAME:TABISH KHAN");  
  
        List<String> pro = new ArrayList<>();  
  
        pro.add("p1");  
        pro.add("p2");  
        pro.add("p3");  
        pro.add("p4");  
        pro.add("p5");  
  
        ListIterator<String> fi = pro.listIterator();  
  
        System.out.println("Forward iteration:");  
  
        while (fi.hasNext()) {  
            System.out.println(fi.next());  
        }  
  
        ListIterator<String> bi = pro.listIterator(pro.size());  
  
        System.out.println("\nBackward iteration:");  
  
        while (bi.hasPrevious()) {  
            System.out.println(bi.previous());  
        }  
    }  
}
```

```
}
```

Output :-

The screenshot shows the Eclipse IDE's Console view. The output window title is "Console X". The console displays the following text:

```
17     }
18     i<-->iterator<String> hi = pno.iterator(pno.size());
<terminated> prac2b [Java Application] C:\Users\student\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 10:04:57 am - 10:04:57 am elapsed: 0:00:00.143) [pid: 16656]
ROLLNO:25MCA-33
NAME:TABISH KHAN
Forward iteration:
p1
p2
p3
p4
p5

Backward iteration:
p5
p4
p3
p2
p1
```

The status bar at the bottom of the console window shows "Activate Windows" and "Go to Settings to activate Windows." It also displays "Writable", "Smart Insert", and a ratio of "1:1:0".

PRACTICAL NO. 3**Aim :-**

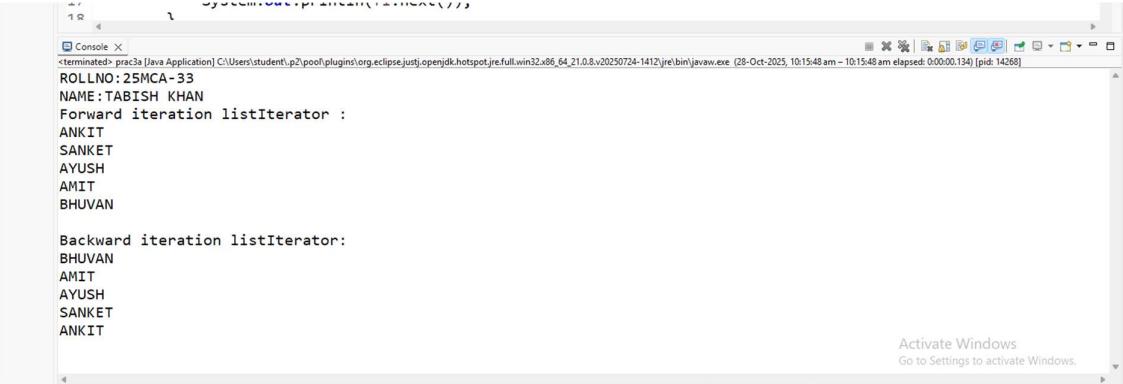
3.1 Write a Java program to create a Set containing list of items of type String and print the items in the list using Iterator interface. Also print the list in reverse / backward direction.

Code :-

```
package prac3;  
import java.util.*;  
  
public class prac3a {  
  
    public static void main(String[] args) {  
  
        System.out.println("ROLLNO:25MCA-33");  
  
        System.out.println("NAME:TABISH KHAN");  
  
        LinkedList<String> proList = new LinkedList<>();  
  
        proList.add("ANKIT");  
  
        proList.add("SANKET");  
  
        proList.add("AYUSH");  
  
        proList.add("AMIT");  
  
        proList.add("BHUVAN");  
  
        ListIterator<String> fi = proList.listIterator();  
  
        System.out.println("Forward iteration listIterator :");  
  
        while (fi.hasNext()) {  
  
            System.out.println(fi.next());  
  
        }  
  
        ListIterator<String> bi = proList.listIterator(proList.size());  
  
        System.out.println("\nBackward iteration listIterator:");  
  
        while (bi.hasPrevious()) {  
  
            System.out.println(bi.previous());  
  
        }  
    }  
}
```

```
}
```

```
}
```

Output :-

The screenshot shows the Eclipse IDE's Console view. The output window title is "Console". The log message starts with the path "<terminated>: prac3a [Java Application] C:\Users\student\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32\x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 10:15:48 am – 10:15:49 am elapsed: 0:00:00.134) [pid: 14268]". The log then displays two sets of names: one for forward iteration and one for backward iteration of a list iterator.

```
ROLLNO: 25MCA-33
NAME: TABISH KHAN
Forward iteration listIterator :
ANKIT
SANKET
AYUSH
AMIT
BHUVAN

Backward iteration listIterator:
BHUVAN
AMIT
AYUSH
SANKET
ANKIT
```

Aim :-

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

Add items in the set.

Insert items of one set in to other set.

Remove items from the set

Search the specified item in the set**Code :-**

```
package prac3;  
import java.util.*;  
  
public class prac3b {  
  
    public static void main(String[] args) {  
  
        System.out.println("ROLLNO:25MCA-33");  
  
        System.out.println("NAME:TABISH KHAN");  
  
        Set<String> set1 = new HashSet<>();  
  
        set1.add("ANKIT");  
  
        set1.add("SANKET");  
  
        set1.add("AYUSH");  
  
        set1.add("AMIT");  
  
        set1.add("BHUVAN");  
  
        System.out.println("Set after adding elements: " + set1);  
  
        Set<String> set2 = new HashSet<>();  
  
        set2.add("RAVI");  
  
        set2.add("MANOJ");  
  
        set2.add("VIKRAM");  
  
        set1.addAll(set2);  
  
        System.out.println("Set1 after adding all elements of Set2: " + set1);  
  
        set1.remove("AMIT");
```

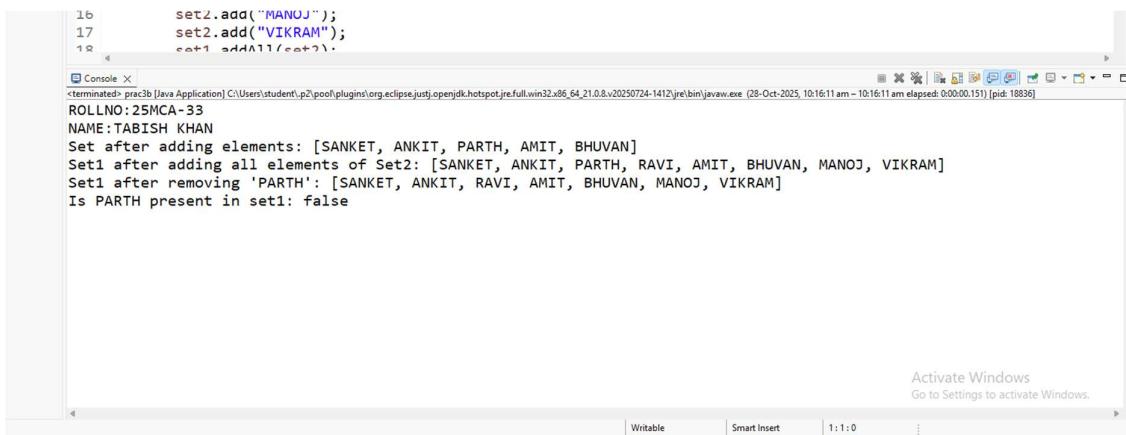
```
System.out.println("Set1 after removing 'AMIT': " + set1);

boolean isPresent = set1.contains("AYUSH");

System.out.println("Is AYUSH present in set1: " + isPresent);

}

}
```

Output :-

The screenshot shows the Eclipse IDE's Console view. The code in the editor is:

```
16      set2.add("MANOJ");
17      set2.add("VIKRAM");
18      set1.addAll(set2);
19
20      System.out.println("Set after adding elements: " + set1);
21      System.out.println("Set1 after adding all elements of Set2: " + set1);
22      System.out.println("Set1 after removing 'PARTH': " + set1);
23      boolean isPresent = set1.contains("AYUSH");
24      System.out.println("Is AYUSH present in set1: " + isPresent);
```

The console output is:

```
terminated> prac3b [Java Application] C:\Users\student1.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 10:16:11 am - 10:16:11 am elapsed: 0:00:00.151) [pid: 18836]
ROLLNO:25MCA-33
NAME:TABISH KHAN
Set after adding elements: [SANKET, ANKIT, PARTH, AMIT, BHUVAN]
Set1 after adding all elements of Set2: [SANKET, ANKIT, PARTH, RAVI, AMIT, BHUVAN, MANOJ, VIKRAM]
Set1 after removing 'PARTH': [SANKET, ANKIT, RAVI, AMIT, BHUVAN, MANOJ, VIKRAM]
Is AYUSH present in set1: false
```

At the bottom right of the console window, there is a message: "Activate Windows Go to Settings to activate Windows."

PRACTICAL NO. 4

Aim :- Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations:

Add items in the map.

Remove items from the map

Search specific key from the map

Get value of the specified key

Insert map elements of one map in to other map.

Print all keys and values of the map.

Code:-

```
package prac4;  
import java.util.*;  
  
public class prac4a {  
  
    public static void main(String[] args) {  
  
        System.out.println("ROLL NO:25MCA-33");  
        System.out.println("NAME:TABISH KHAN");  
  
        Map<Integer,String>map1=new HashMap<>();  
        map1.put(1,"Aniket");  
        map1.put(2,"Suresh");  
        map1.put(3,"Tabish");  
  
        System.out.println("map1: "+map1);  
  
        Map<Integer,String>map2=new HashMap<>();  
        map2.put(4,"Amit");  
        map2.put(5,"Bhuvan");  
        map2.put(6,"Max");  
  
        System.out.println("map2: "+map2);  
        map1.putAll(map2);  
  
        System.out.println("map1 after adding map2 in map1: "+map1);  
    }  
}
```

```
map1.remove(5);
System.out.println("map1 after removing :" + map1);
boolean isPresent = map1.containsKey(1);
System.out.println("is apple present: " + isPresent);
String value = map1.get(2);
System.out.println("Value for key 2 is:" + value);
for (Map.Entry<Integer, String> entry : map1.entrySet())
{
    System.out.println("key: " + entry.getKey() + " values: " + entry.getValue());
}
}
```

Output :-

```
ROLL NO:25MCA-33
NAME:TABISH KHAN
map1: {1=Aniket, 2=Suresh, 3=Tabish}
map2: {4=Amit, 5=Bhuvan, 6=Max}
map1 after adding map2 in map1: {1=Aniket, 2=Suresh, 3=Tabish, 4=Amit, 5=Bhuvan, 6=Max}
map1 after removing :{1=Aniket, 2=Suresh, 3=Tabish, 4=Amit, 6=Max}
is apple present: false
Value for key 2 is:Suresh
key: 1 values: Aniket
key: 2 values: Suresh
key: 3 values: Tabish
key: 4 values: Amit
key: 6 values: Max
```

PRACTICAL NO. 5**Aim :-****5.1 Write a program using Lambda Expression to print “Hello World”.****Code :-**

```
package prac5;

public class prac5a {

    public static void main(String[] args) {

        Runnable helloworld =()->System.out.println("Hello World");

        Runnable Name =()->System.out.println("Tabish Khan");

        Runnable Rollno =()->System.out.println("25MCA-33");

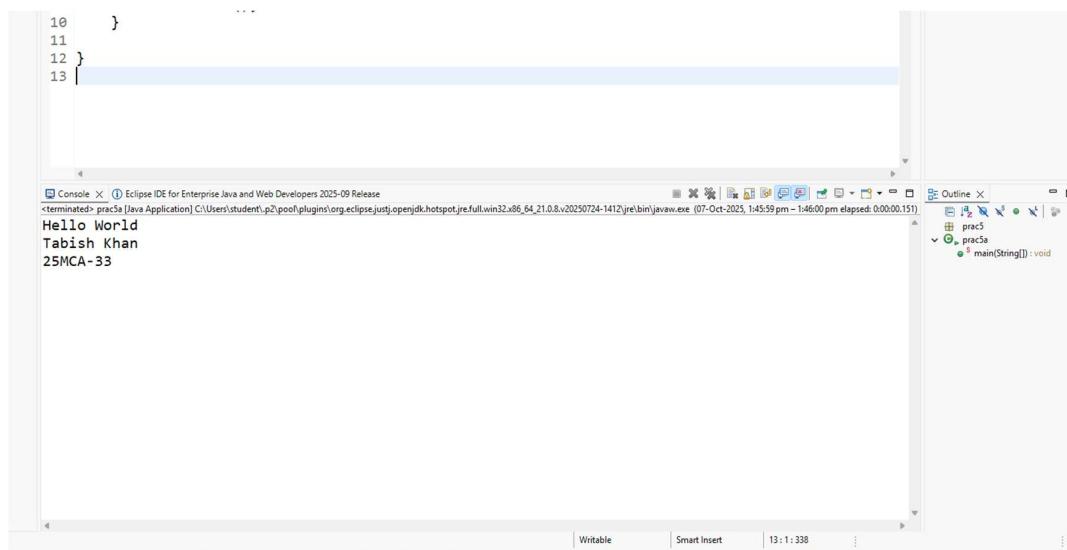
        helloworld.run();

        Name.run();

        Rollno.run();

    }

}
```

Output :-

Aim :-

5.2 Write a program using Lambda Expression to print names with single parameters.

Code :-

```
package prac5;  
  
import java.util.*;  
  
public class prac5b {  
  
    public static void main(String[] args) {  
  
        System.out.println("Rollno:25MCA-33");  
  
        System.out.println("Name: Khan Tabish Mujeeb");  
  
        List<String>names=Arrays.asList("Tabish","Ahsan","Majid");  
  
        names.forEach(name->System.out.println("Hello "+name+"!"));  
  
    }  
}
```

Output :-

The screenshot shows the Eclipse IDE interface with the following details:

- Code Editor:** Displays Java code in a file named `prac3b`. The code prints "Hello" followed by each name from a list: "Tabish", "Ahsan", and "Majid".
- Console View:** Shows the terminal output of the executed code. It includes:
 - Environment information: Console X Eclipse IDE for Enterprise Java and Web Developers 2025-09 Release, terminated: `prac3b [Java Application] C:\Users\student\.p2\pool\plugins\org.eclipse.jdt.core\full\win32\x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe`.
 - Execution results:
 - Rollno:25MCA-33
 - Name: Khan Tabish Mujeeb
 - Hello Tabish!
 - Hello Ahsan!
 - Hello Majid!
- Outline View:** Shows the project structure with `prac3b` selected, containing the `main(String[])` method.

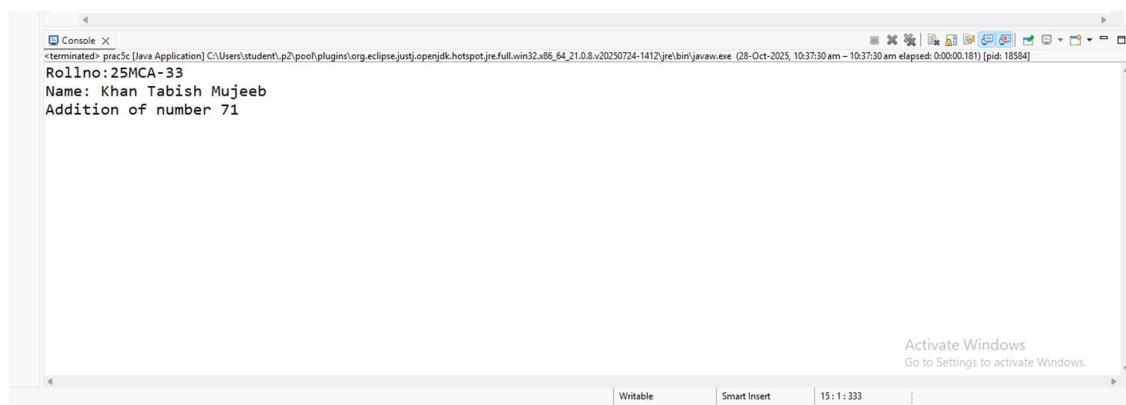
Aim :-

5.3 Write a program using Lambda Expression with multiple parameters to add two numbers.

```
package prac5;

interface A{
    int add(int i,int j);
}

public class prac5c {
    public static void main(String[] args) {
        A num=(i,j)->i+j;
        int result = num.add(15,56);
        System.out.println("Rollno:25MCA-33");
        System.out.println("Name: Khan Tabish Mujeeb");
        System.out.println("Addition of number "+result);
    }
}
```

Output :-

The screenshot shows a Windows command-line interface window titled "Console X". The output of the Java application "prac5c" is displayed, which includes the roll number, name, and the result of the addition. The console window has standard Windows-style icons at the top and a status bar at the bottom indicating "Activate Windows" and the current time "15:1:333".

```
<terminated> prac5c [Java Application] C:\Users\student\p2pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.v86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 10:37:30 am - 10:37:30 am elapsed: 0:00:00.181) [pid: 18584]
Rollno:25MCA-33
Name: Khan Tabish Mujeeb
Addition of number 71
```

Aim :-

5.4 Write a program using Lambda Expression to calculate the following :

- 1.Convert Fahrenheit to Celsius.**
- 2.Convert Kilometers to Miles.**

Code :-

```
package prac5;

interface Converter{

    double convert(double input);

}

public class pracE {

    public static void main(String[] args) {

        System.out.println("ROLLNO:25MCA-33");

        System.out.println("NAME:TABISH KHAN");

        Converter a=f->(f-32)*5/9;

        double celcius=a.convert(100);

        System.out.println("100 Degree fahrenheit is "+celcius+" Degree celcius");

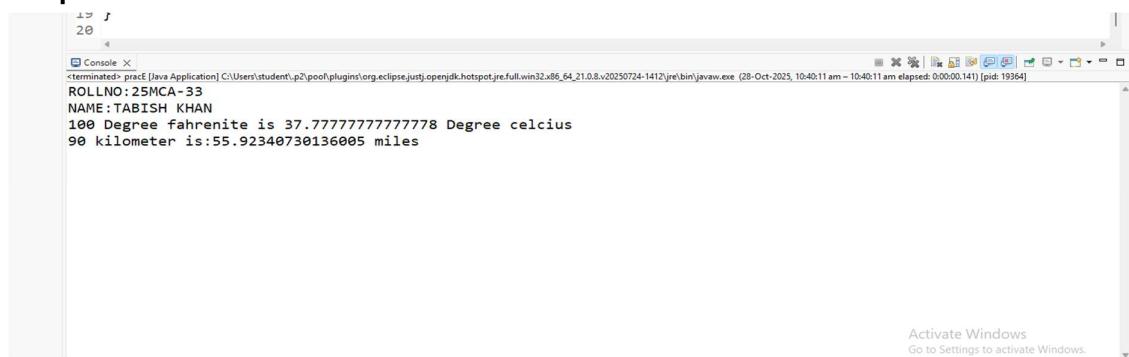
        Converter b=km->km/1.609344;

        double miles=b.convert(90);

        System.out.println("90 kilometer is:"+miles+" miles");

    }

}
```

Output :-


```
20
ROL NO : 25MCA - 33
NAME : TABISH KHAN
100 Degree fahrenheit is 37.777777777778 Degree celcius
90 kilometer is : 55.92340730136005 miles
```

Aim :-

5.5 Write a program using Lambda Expression with or without return keyword.

Code :-

```
package prac5;

interface Calculator{

    int calculate(int i,int j);

}

public class prac5e {

    public static void main(String[] args) {

        System.out.println("ROLLNO:25MCA-33");

        System.out.println("NAME:TABISH KHAN");

        Calculator add=(i,j)->i+j;

        Calculator subtract=(i,j)->{

            return i=j;

        };

        int sum=add.calculate(10,8);

        int diffrence=subtract.calculate(78,45);

        System.out.println("Without return keyword addition:" +sum);

        System.out.println("with return keyword diffrence:"+diffrence);

    }

}
```

Output :-

The screenshot shows the Eclipse IDE interface. On the left, there is a code editor window with Java code. On the right, there is a 'Console' window displaying the program's output. The output text is:

```
<terminated>: prac5e [Java Application] C:\Users\student\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (28-Oct-2025, 10:48:00 am – 10:48:01 am elapsed: 0:00:00.167)
ROLLNO:25MCA-33
NAME:TABISH KHAN
Without return keyword addition:18
with return keyword diffrence:45
```

The code editor shows a class named 'prac5e' with a main method and a calculate method. The 'Outline' view on the right shows the class structure.

Aim :-**5.6 Write a program using Lambda Expression to concatenate two Strings:****Code :-**

```
package prac5;

public interface Concatentor{

    String concatenate(String s1, String s2);

}

public class prac5f{

    public static void main(String[] args){

        String str = "Khan";

        String str1 = " Tabish";

        Concatentor s = (s1, s2) -> s1 + s2;

        String res = s.concatenate(str, str1);

        System.out.println("The concatenation string is " + res);

    }

}
```

Output :-

PRACTICAL NO. 6**Aim :-****6.2 Write a JSP page to display the Registration form****Code :-**

registration.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>
<head>
<meta charset="UTF-8">
<title>Complete Registration Form</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #e9f0f5;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
.container {
    width: 600px;
    background-color: white;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 0 15px rgba(0,0,0,0.2);
}
h2 {
```

```
text-align: center;  
color: #333;  
}  
  
form {  
display: grid;  
grid-template-columns: 1fr 1fr;  
grid-gap: 15px;  
}  
  
label {  
font-weight: bold;  
}  
  
input, select, textarea {  
width: 100%;  
padding: 8px;  
border: 1px solid #ccc;  
border-radius: 5px;  
}  
  
.full-width {  
grid-column: span 2;  
}  
  
.buttons {  
grid-column: span 2;  
text-align: center;  
}  
  
input[type="submit"], input[type="reset"] {  
width: 120px;  
background-color: #007bff;  
color: white;
```

```
border: none;  
cursor: pointer;  
padding: 10px;  
border-radius: 5px;  
margin: 10px;  
}  
  
input[type="submit"]:hover, input[type="reset"]:hover {  
background-color: #0056b3;  
}  
</style>  
</head>  
<body>  
<div class="container">  
<h2>Registration Form</h2>  
<form action="registerUser.jsp" method="post">  
    <label for="fullname">Full Name:</label>  
    <input type="text" id="fullname" name="fullname" required>  
    <label for="username">Username:</label>  
    <input type="text" id="username" name="username" required>  
    <label for="password">Password:</label>  
    <input type="password" id="password" name="password" required>  
    <label for="confirm">Confirm Password:</label>  
    <input type="password" id="confirm" name="confirm" required>  
    <label for="gender">Gender:</label>  
    <select id="gender" name="gender" required>  
        <option value="">-- Select --</option>  
        <option>Male</option>  
        <option>Female</option>
```

```
<option>Other</option>
</select>

<label for="dob">Date of Birth:</label>
<input type="date" id="dob" name="dob">

<label for="email">Email:</label>
<input type="email" id="email" name="email" required>

<label for="phone">Phone:</label>
<input type="text" id="phone" name="phone" required>

<label for="country">Country:</label>
<select id="country" name="country">
    <option>India</option>
    <option>United States</option>
    <option>United Kingdom</option>
    <option>Canada</option>
    <option>Australia</option>
    <option>Other</option>
</select>

<label for="state">State:</label>
<input type="text" id="state" name="state">

<label>Hobbies:</label>
<div>
    <input type="checkbox" name="hobby" value="Reading"> Reading
    <input type="checkbox" name="hobby" value="Traveling"> Traveling
    <input type="checkbox" name="hobby" value="Music"> Music
    <input type="checkbox" name="hobby" value="Sports"> Sports
</div>

<label for="address" class="full-width">Address:</label>
<textarea id="address" name="address" rows="3" class="full-width"></textarea>
```

```
<div class="buttons">
    <input type="submit" value="Register">
    <input type="reset" value="Reset">
</div>
</form>
</div>
</body>
</html>
```

registerUser.jsp

```
<%@ page import="java.sql.*;java.util.*" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <title>Registration Result</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #e9f0f5;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .message-box {
            background: white;
            padding: 30px;
        }
    </style>
</head>
<body>
    <div class="message-box">
        <form>
            <div class="buttons">
                <input type="submit" value="Register">
                <input type="reset" value="Reset">
            </div>
        </form>
    </div>
</body>
</html>
```

```
border-radius: 10px;  
box-shadow: 0 0 15px rgba(0,0,0,0.2);  
width: 450px;  
text-align: center;  
}  
  
h2 { color: #007bff; }  
  
p { font-size: 16px; }  
  
a {  
display: inline-block;  
margin-top: 15px;  
padding: 10px 15px;  
background-color: #007bff;  
color: white;  
text-decoration: none;  
border-radius: 5px;  
}  
  
a:hover { background-color: #0056b3; }  
  
</style>  
</head>  
<body>  
<div class="message-box">  
  <%  
    request.setCharacterEncoding("UTF-8");  
    String fullname = request.getParameter("fullname");  
    String username = request.getParameter("username");  
    String password = request.getParameter("password");  
    String confirm = request.getParameter("confirm");  
    String gender = request.getParameter("gender");
```

```

String dob    = request.getParameter("dob");
String email  = request.getParameter("email");
String phone   = request.getParameter("phone");
String country = request.getParameter("country");
String state   = request.getParameter("state");
String[] hobbies = request.getParameterValues("hobby");
String address = request.getParameter("address");
if (password == null) password = "";
if (confirm == null) confirm = "";
String hobbyList = "";
if (hobbies != null && hobbies.length > 0) {
    hobbyList = String.join(", ", hobbies);
}
if (!password.equals(confirm)) {
    out.println("<h2 style='color:red;'>Password and Confirm Password do not
match!</h2>");
} else {
    Connection conn = null;
    PreparedStatement ps = null;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url =
"jdbc:mysql://localhost:3306/penta?useSSL=false&serverTimezone=UTC";
        String dbUser = "root"; // your MySQL username
        String dbPass = "gnims"; // your MySQL password
        conn = DriverManager.getConnection(url, dbUser, dbPass);
        String sql = "INSERT INTO users (fullname, username, password, gender, dob, email,
phone, country, state, hobbies, address) " +
"VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
    }
}

```

```
ps = conn.prepareStatement(sql);

ps.setString(1, fullname);

ps.setString(2, username);

ps.setString(3, password);

ps.setString(4, gender);

if (dob != null && !dob.isEmpty()) {

    ps.setDate(5, java.sql.Date.valueOf(dob));

} else {

    ps.setNull(5, java.sql.Types.DATE);

}

ps.setString(6, email);

ps.setString(7, phone);

ps.setString(8, country);

ps.setString(9, state);

ps.setString(10, hobbyList);

ps.setString(11, address);

int rows = ps.executeUpdate();

if (rows > 0) {

    out.println("<h2>Registration Successful!</h2>");

    out.println("<p>Welcome, " + fullname + ".</p>");

} else {

    out.println("<h2 style='color:red;'>Registration Failed.</h2>");

}

} catch (ClassNotFoundException e) {

    out.println("<h2 style='color:red;'>MySQL JDBC Driver not found! Please add it to Tomcat/lib.</h2>");

} catch (SQLException e) {

    out.println("<h2 style='color:red;'>Database Error: " + e.getMessage() + "</h2>");
```

```

} catch (Exception e) {
    out.println("<h2 style='color:red;'>Unexpected Error: " + e.getMessage() + "</h2>");
}

} finally {

try { if (ps != null) ps.close(); } catch (SQLException ignored) {}

try { if (conn != null) conn.close(); } catch (SQLException ignored) {}

}

%>

<a href="registration.jsp">Go Back</a>
</div>
</body>
</html>

```

Sql query:-

```

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    fullname VARCHAR(100) NOT NULL,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    gender VARCHAR(10),
    dob DATE,
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20),
    country VARCHAR(50),
    state VARCHAR(50),
    hobbies VARCHAR(255),
    address TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Output :-

The registration form fields are as follows:

- Full Name:** [Text Input]
- Username:** [Text Input]
- Password:** [Text Input]
- Confirm Password:** [Text Input]
- Gender:** [Select Box] -- Select --
- Date of Birth:** [Text Input] dd-mm-yyyy
- Email:** [Text Input]
- Phone:** [Text Input]
- Country:** [Select Box] India
- State:** [Text Input]
- Hobbies:**
 - Reading
 - Traveling
 - Music
 - Sports
- Address:** [Text Area]

Buttons at the bottom: Register (blue) and Reset.

Registration Successful!
Welcome, Tabish khan.
[Go Back](#)

DATABASE :-

```
mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | fullname | username | password | gender | dob | email | phone | country | state | hobbies | address | created_at |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | amit | admin | 123456 | Male | 2025-10-31 | chetanmaurya02@gmail.com | 1234567890 | India | rfgere | Reading | dsfewtrf4ew | 2025-11-10 10:25:14 |
| 4 | Tabish khan | tk | 123456 | Male | 2007-03-07 | AB@gmail.com | 1234567890 | India | up | Reading, Traveling | Mumbai | 2025-11-10 11:00:14 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Aim :-

6.3 Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table.

Code :-

Studentmaster.jsp

```
<%@ page import="java.sql.*" %>

<html>
<head>
    <title>Student Master</title>
    <link
        href="https://fonts.googleapis.com/css2?family=Baloo+2:wght@600&display=swap"
        rel="stylesheet">
    <style>
        body {
            font-family: 'Baloo 2', cursive;
            margin: 30px;
            background: url('https://images.pexels.com/photos/691668/pexels-photo-691668.jpeg') no-repeat center center fixed;
            background-size: cover;
            color: #222;
        }
        h2, h3 {
            color: #ff3030;
            text-shadow: 3px 3px yellow;
            font-size: 40px;
        }
        table {
            border-collapse: collapse;
            width: 75%;
            margin-top: 20px;
        }
    </style>

```

```
background: rgba(255,255,255,0.85);  
border-radius: 15px;  
overflow: hidden;  
box-shadow: 0px 0px 15px 4px rgba(0,0,0,0.4);  
}  
  
th {  
background-color: #ff0000;  
color: white;  
font-size: 20px;  
padding: 12px;  
border-bottom: 4px solid black;  
}  
  
td {  
border: 2px solid #000;  
padding: 10px;  
font-size: 18px;  
}  
  
input[type=text] {  
width: 200px;  
padding: 8px;  
border: 3px solid #000;  
border-radius: 10px;  
font-size: 16px;  
background: #fff8d6;  
}  
  
input[type=submit] {  
padding: 10px 20px;  
font-size: 18px;
```

```
border-radius: 12px;  
cursor: pointer;  
font-weight: bold;  
border: 3px solid #000;  
transition: 0.2s;  
background: #ffcd00;  
}  
  
input[type=submit]:hover {  
background: #ffe066;  
transform: scale(1.1);  
}  
  
input[value="Delete"] {  
background: #ff1e1e;  
color: white;  
}  
  
input[value="Delete"]:hover {  
background: #ff5252;  
}  
  
input[value="Edit"] {  
background: #1e90ff;  
color: white;  
}  
  
input[value="Edit"]:hover {  
background: #65b5ff;  
}  
  
input[value="Update"] {  
background: #ffa500;  
color: black;
```

```
        }

    .msg {
        margin-top: 15px;
        font-weight: bold;
        background: yellow;
        width: fit-content;
        padding: 10px 20px;
        border: 3px solid black;
        border-radius: 10px;
        font-size: 22px;
    }

}

</style>

</head>

<body>

<h2>Student Master</h2>

<%
    String url = "jdbc:mysql://localhost:3306/penta";
    String user = "root";
    String pass = "gnims";
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String message = "";
    String editRoll = "";
    String editName = "";
    String editSem = "";
    String editCourse = "";
    boolean isEdit = false;
}
```

```
try {  
    Class.forName("com.mysql.cj.jdbc.Driver");  
    con = DriverManager.getConnection(url, user, pass);  
    String action = request.getParameter("action");  
    if ("Add".equals(action)) {  
        int roll = Integer.parseInt(request.getParameter("rollno"));  
        String name = request.getParameter("name");  
        int sem = Integer.parseInt(request.getParameter("semester"));  
        String course = request.getParameter("course");  
        ps = con.prepareStatement("INSERT INTO studentmaster VALUES (?, ?, ?, ?)");  
        ps.setInt(1, roll);  
        ps.setString(2, name);  
        ps.setInt(3, sem);  
        ps.setString(4, course);  
        ps.executeUpdate();  
        message = "Student Added Successfully!";  
    } else if ("Delete".equals(action)) {  
        int roll = Integer.parseInt(request.getParameter("rollno"));  
        ps = con.prepareStatement("DELETE FROM studentmaster WHERE rollno=?");  
        ps.setInt(1, roll);  
        int rows = ps.executeUpdate();  
        if (rows > 0)  
            message = "Student Deleted Successfully!";  
        else  
            message = "No student found with Roll No " + roll;  
    } else if ("Edit".equals(action)) {  
        int roll = Integer.parseInt(request.getParameter("rollno"));  
        ps = con.prepareStatement("SELECT * FROM studentmaster WHERE rollno=?");  
    }  
}
```

```
ps.setInt(1, roll);

rs = ps.executeQuery();

if (rs.next()) {

    editRoll = rs.getString("rollno");

    editName = rs.getString("name");

    editSem = rs.getString("semester");

    editCourse = rs.getString("course");

    isEdit = true;

}

} else if ("Update".equals(action)) {

    int roll = Integer.parseInt(request.getParameter("rollno"));

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

    int sem = Integer.parseInt(request.getParameter("semester"));

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

    ps = con.prepareStatement("UPDATE studentmaster SET name=?, semester=?, course=? WHERE rollno=?");

    ps.setString(1, name);

    ps.setInt(2, sem);

    ps.setString(3, course);

    ps.setInt(4, roll);

    ps.executeUpdate();

    message = "Student Updated Successfully!";

}

} catch (Exception e) {

    message = "Error: " + e.getMessage();

}

%>

<form method="post" action="studentmaster.jsp">

<table>
```

```
<tr>
<td>Roll No:</td>
<td><input type="text" name="rollno" value="<% if (isEdit) { editRoll } else { "" } %>" required></td>
</tr>
<tr>
<td>Name:</td>
<td><input type="text" name="name" value="<% if (isEdit) { editName } else { "" } %>" required></td>
</tr>
<tr>
<td>Semester:</td>
<td><input type="text" name="semester" value="<% if (isEdit) { editSem } else { "" } %>" required></td>
</tr>
<tr>
<td>Course:</td>
<td><input type="text" name="course" value="<% if (isEdit) { editCourse } else { "" } %>" required></td>
</tr>
</table>
<br>
<% if (isEdit) { %>
<input type="submit" name="action" value="Update">
<% } else { %>
<input type="submit" name="action" value="Add">
<% } %>
</form>
<div class="msg"><% message %></div>
<h3>All Student Records</h3>
```

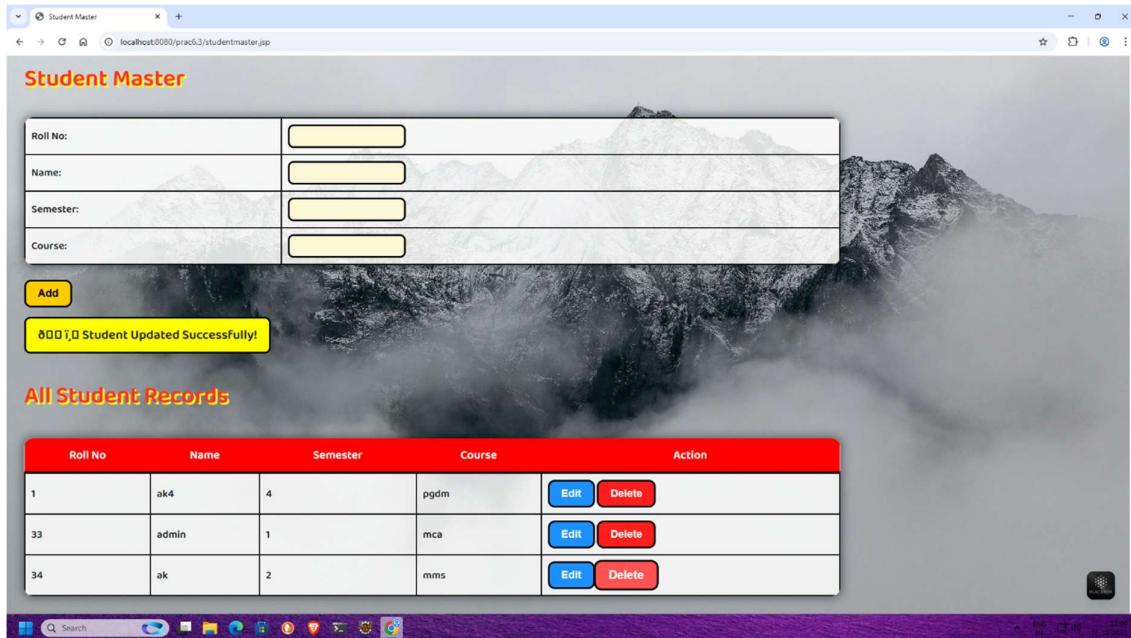
```
<table>
<tr>
    <th>Roll No</th>
    <th>Name</th>
    <th>Semester</th>
    <th>Course</th>
    <th>Action</th>
</tr>
<%
try {
    Statement st = con.createStatement();
    rs = st.executeQuery("SELECT * FROM studentmaster ORDER BY rollno");
    while (rs.next()) {
%>
<tr>
    <td><%= rs.getInt("rollno") %></td>
    <td><%= rs.getString("name") %></td>
    <td><%= rs.getInt("semester") %></td>
    <td><%= rs.getString("course") %></td>
    <td>
        <form method="post" action="studentmaster.jsp" style="display:inline;">
            <input type="hidden" name="rollno" value=<%= rs.getInt("rollno") %>>
            <input type="submit" name="action" value="Edit">
        </form>
        <form method="post" action="studentmaster.jsp" style="display:inline;">
            <input type="hidden" name="rollno" value=<%= rs.getInt("rollno") %>>
            <input type="submit" name="action" value="Delete">
        </form>
    </td>

```

```
</td>
</tr>
<%
}
} catch (Exception e) {
    out.println("<tr><td colspan='5'>Error: " + e.getMessage() + "</td></tr>");
} finally {
    if (rs != null) rs.close();
    if (ps != null) ps.close();
    if (con != null) con.close();
}
%>
</table>
</body>
</html>
```

Sql queries:-

```
CREATE TABLE studentmaster (
    rollno INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    semester INT NOT NULL,
    course VARCHAR(100) NOT NULL
);
```

Output :-**DATABASE:-**

```
mysql> select * from studentmaster;
+-----+-----+-----+-----+
| rollno | name | semester | course |
+-----+-----+-----+-----+
|     1 | ak4  |         4 | pgdm   |
|    33 | admin |         1 | mca    |
|    34 | ak   |         2 | mms    |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Aim :-

6.4 Design Loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount. Display the payment amount for each loan and then list the loan balance and interest paid for each payment over the term of the loan for the following time period and interest rate:

- a)1 to 7 year at 5.35%
- b)8 to 15 year at 5.5%
- c)16 to 30 year at 5.75%

Code :-

Index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>
<head>
    <title>Loan Calculator</title>
    <style>
        body { font-family: Arial; background: #f4f4f4; text-align: center; padding: 50px; }
        form { background: white; display: inline-block; padding: 20px; border-radius: 8px; }
        input { margin: 10px; padding: 8px; }
        button { padding: 10px 20px; background-color: #4CAF50; color: white; border: none; border-radius: 4px; }
    </style>
</head>
<body>
    <h2>Loan Calculator</h2>
    <form action="calculate.jsp" method="post">
        <label>Principal Loan Amount:</label><br>
        <input type="number" name="principal" step="0.01" required><br>
        <label>Period of Time (in years):</label><br>
        <input type="number" name="years" required><br>
```

```
<button type="submit">Calculate</button>  
</form>  
</body>  
</html>
```

Calculate.jsp

```
<%@ page import="java.text.DecimalFormat" %>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Loan Calculation Result</title>  
    <style>  
        body { font-family: Arial; background: #f9f9f9; text-align: center; padding: 40px; }  
        table { margin: auto; border-collapse: collapse; width: 80%; }  
        th, td { border: 1px solid #ccc; padding: 10px; text-align: right; }  
        th { background-color: #f2f2f2; }  
    </style>  
</head>  
<body>  
<%  
    double principal = Double.parseDouble(request.getParameter("principal"));  
    int years = Integer.parseInt(request.getParameter("years"));  
    double annualRate = 0.0;  
    if (years >= 1 && years <= 7) {  
        annualRate = 5.35;  
    } else if (years >= 8 && years <= 15) {
```

```
annualRate = 5.5;

} else if (years >= 16 && years <= 30) {

    annualRate = 5.75;

} else {

    out.println("<h3>Invalid time period. Please enter between 1 and 30 years.</h3>");

    return;
}

double monthlyRate = annualRate / 100 / 12;

int months = years * 12;

double monthlyPayment = principal * monthlyRate / (1 - Math.pow(1 + monthlyRate, -months));

DecimalFormat df = new DecimalFormat("#,##0.00");

out.println("<h2>Loan Calculation Result</h2>");

out.println("<p><b>Principal:</b> $" + df.format(principal) + "</p>");

out.println("<p><b>Years:</b> " + years + "</p>");

out.println("<p><b>Annual Interest Rate:</b> " + annualRate + "%</p>");

out.println("<p><b>Monthly Payment:</b> $" + df.format(monthlyPayment) + "</p>");

%>

<table>

<tr>

    <th>Payment No.</th>

    <th>Interest Paid</th>

    <th>Principal Paid</th>

    <th>Remaining Balance</th>

</tr>

<%>

double balance = principal;

for (int i = 1; i <= months; i++) {

    double interest = balance * monthlyRate;
```

```

double principalPaid = monthlyPayment - interest;
balance -= principalPaid;
if (balance < 0) balance = 0;
out.println("<tr>");
out.println("<td>" + i + "</td>");
out.println("<td>$" + df.format(interest) + "</td>");
out.println("<td>$" + df.format(principalPaid) + "</td>");
out.println("<td>$" + df.format(balance) + "</td>");
out.println("</tr>");
}

%>
</table>

<p><a href="index.jsp">Back to Calculator</a></p>
</body>
</html>

```

Output :-

The screenshot shows a web browser window with the title "JSP Declaration, Scriptlet, Directive Example" and the sub-page "Loan Calculation Result". The URL in the address bar is "localhost:8080/pracs/calculate.jsp". The main content is titled "Loan Calculation Result" and provides the following loan details:

- Principal: \$50,000.00
- Years: 1
- Annual Interest Rate: 5.35%
- Monthly Payment: \$4,288.40

Below these details is a table showing the payment schedule over 12 months:

Payment No.	Interest Paid	Principal Paid	Remaining Balance
1	\$222.92	\$4,065.48	\$45,934.52
2	\$204.79	\$4,083.61	\$41,850.91
3	\$186.59	\$4,101.81	\$37,749.10
4	\$168.30	\$4,120.10	\$33,628.00
5	\$149.93	\$4,138.47	\$29,490.53
6	\$131.48	\$4,156.92	\$25,333.61
7	\$112.95	\$4,175.45	\$21,158.16
8	\$94.33	\$4,194.07	\$16,964.09
9	\$75.63	\$4,212.77	\$12,751.33
10	\$56.85	\$4,231.55	\$8,519.78
11	\$37.98	\$4,250.41	\$4,269.36
12	\$19.03	\$4,269.36	\$0.00

At the bottom of the table is a link "Back to Calculator".

Aim :-

6.5 Write a program using JSP that displays a webpage consisting an Application form for change of Study Center which can be filled by any student who wants to change his/her study center. Make necessary assumptions

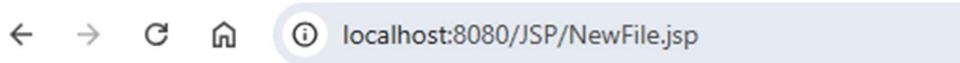
Code :-

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %> <!DOCTYPE html>

<html>
<head>
<title>Change Study Center Application</title>
</head>
<body>
<%
String studentName = request.getParameter("studentName");
String currentCenter = request.getParameter("currentCenter");
String newCenter = request.getParameter("newCenter");
%>
<h1>Change Study Center Application</h1>
<form method="post">
Student Name: <input type="text" name="studentName" value=<%= (studentName != null ? studentName : "") %>" required><br><br>
Current Study Center: <input type="text" name="currentCenter" value=<%= (currentCenter != null ? currentCenter : "") %>" required><br><br>
New Study Center: <input type="text" name="newCenter" value=<%= (newCenter != null ? newCenter : "") %>" required><br><br>
<input type="submit" value="Submit">
</form>
<%
if (studentName != null && currentCenter != null && newCenter != null) { %>
<hr>
```

```
<h2>Application Submitted</h2>  
<p>Thank you for submitting your study center change application.</p> <p><b>Student  
Name:</b> <%= studentName %></p> <p><b>Current Center:</b> <%= currentCenter  
%></p> <p><b>New Center:</b> <%= newCenter %></p>  
<%  
}  
%>  
</body>  
</html>
```

Output :-



Change Study Center Application

Student Name:

Current Study Center:

New Study Center:

Application Submitted

Thank you for submitting your study center change application.

Student Name: Tabish Khan

Current Center: Mumbai

New Center: Lucknow

Aim :-

6.6 Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives, expression, header and footer.

Code :-

Main.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>JSP Declaration, Scriptlet, Directive, Expression, Header, and Footer Example</title>
</head>
<body>
<%@ include file="header.jsp" %>
<center>
    <h2>JSP Declaration, Scriptlet, Directive, Expression Example</h2>
    <%-- JSP Declaration: declaring variables and methods --%>
    <%!
        int data = 100;
        double circle(int r) {
            return 3.14 * r * r;
        }
        int rectangle(int l, int b) {
            return l * b;
        }
        int perimeter(int x, int y) {
            int peri = 2 * (x + y);
            return peri;
        }
    %>
</center>
</body>
</html>
```

```

        }

    double triangle(double base, double height) {
        return 0.5 * base * height;
    }

    int square(int side) {
        return side * side;
    }

%>

<%-- JSP Expression: display results dynamically --%>
<p><b>Value of variable:</b> <%= data %></p>
<p><b>Area of Circle (r=15):</b> <%= circle(15) %></p>
<p><b>Area of Rectangle (l=67, b=25):</b> <%= rectangle(67,25) %></p>
<p><b>Perimeter of Rectangle (l=52, b=21):</b> <%= perimeter(52,21) %></p>
<p><b>Area of Triangle (base=10, height=6):</b> <%= triangle(10,6) %></p>
<p><b>Area of Square (side=10):</b> <%= square(10) %></p>
<p><i>Thanks for visiting my page!</i></p>
</center>

<%@ include file="footer.jsp" %>
</body>
</html>
```

Header.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
```

```
<title>Header Section</title>
<%!
    int pageCount = 0;
    void addCount() { pageCount++; }

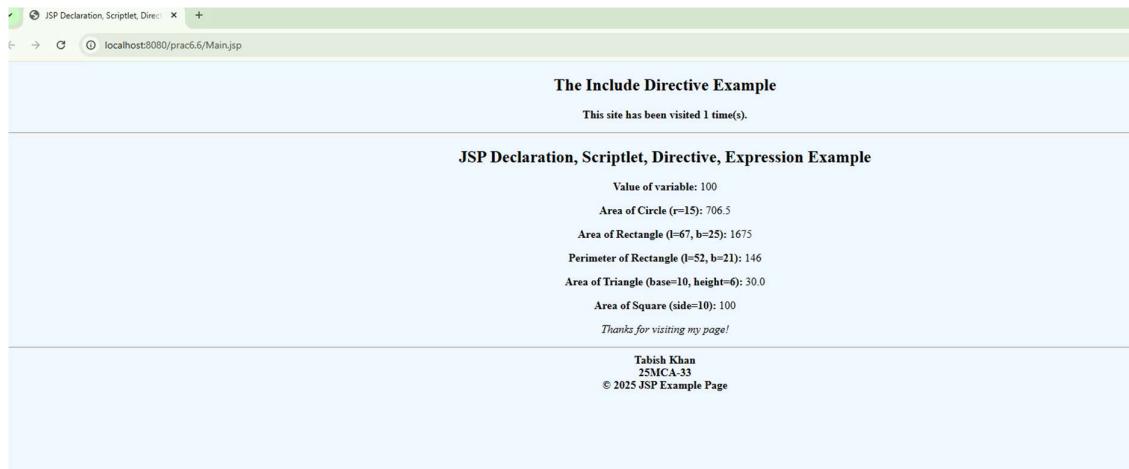
%>
<% addCount(); %>
</head>
<body style="background-color:#f0f8ff;">
<center>
    <h2>The Include Directive Example</h2>
    <p><b>This site has been visited <%= pageCount %> time(s).</b></p>
    <hr>
</center>
</body>
</html>
```

Footer.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="ISO-8859-1">
    <title>Footer Section</title>
</head>
<body>
    <hr>
    <center>
```

```
<b>Tabish Khan</b><br>
<b>25MCA-33</b><br>
<b>© 2025 JSP Example Page</b>
</center>
</body>
</html>
```

Output :-



PRACTICAL NO. 7**Aim :- Assignment based Spring Framework**

1. Write a program to print “Hello World” using spring framework.
2. Write a program to demonstrate dependency injection via setter method.
3. Write a program to demonstrate dependency injection via Constructor.
4. Write a program to demonstrate Autowiring.

Code :-

```
Prac7Application.java

package com.example.demo;

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
@Controller
@ComponentScan(basePackages = "com.example.demo")
public class Prac7Application {
    public static void main(String[] args) {
        SpringApplication.run(Prac7Application.class, args);
    }
    @Bean
    public CommandLineRunner demo(Car car) {
        return args -> {
            car.setModel("Honda");
            Driver dr=new Driver();
        };
    }
}
```

```
        dr.setName("Tabish");

        car.setDriver(dr);

        car.start();

    };

}

}

@RestController

class HelloController {

    private final Car car;

    public HelloController(Car car) {

        this.car = car;

    }

    @GetMapping("/hello")

    public String hello() {

        car.start();

        return car.getModel();

    }

}
```

Car.java

```
package com.example.demo;

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

import org.springframework.stereotype.Component;

@Component

public class Car {

    private String model;

    private Driver driver;

    public String getModel() {
```

```
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public Driver getDriver() {
        return driver;
    }

    @Autowired
    public void setDriver(Driver driver) {
        this.driver = driver;
    }

    public void start() {
        System.out.println("Car model: " + model);
        System.out.println("Driver: " + driver.getName());
        System.out.println("Car is starting...");
    }
}
```

Driver.java

```
package com.example.demo;

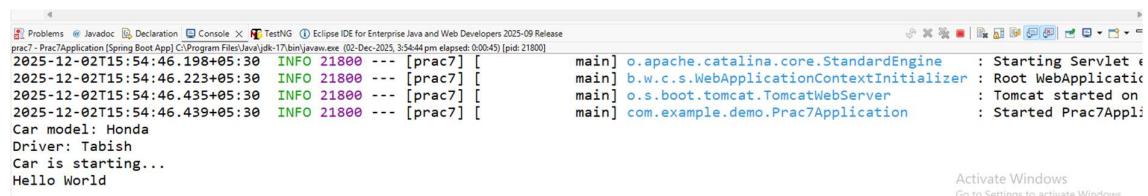
import org.springframework.stereotype.Component;

@Component
public class Driver {

    private String name;

    public String getName() {
        return name;
    }
}
```

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

Output :-

PRACTICAL NO. 8**Aim :- Assignment based Aspect Oriented Programming**

1. Write a program to demonstrate Spring AOP – before advice.
2. Write a program to demonstrate Spring AOP – after advice.
3. Write a program to demonstrate Spring AOP – around advice.
4. Write a program to demonstrate Spring AOP – after returning advice.
5. Write a program to demonstrate Spring AOP – after throwing advice.
6. Write a program to demonstrate Spring AOP – pointcuts.

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>prac-8</artifactId>
  <version>1.0</version>
  <name>Prac-8</name>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.0</version>
  </parent>
  <dependencies>
    <!-- Spring Boot AOP -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-aop</artifactId>
    </dependency>
    <!-- Optional: Spring Web (if needed) -->
  
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
</dependencies>
</project>

Prac8Application.java

package com.example.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import com.example.demo.service.MyService;

@SpringBootApplication
public class Prac8Application implements CommandLineRunner {

    @Autowired
    private MyService service;

    public static void main(String[] args) {
        SpringApplication.run(Prac8Application.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        System.out.println("==> Calling doSomething() ==>");
        service.doSomething();
        System.out.println("-----");
        System.out.println("==> Calling getData() ==>");
        service.getData();
    }
}
```

```
System.out.println("-----");
System.out.println("==> Calling throwError() ==>");
try{
    service.throwError();
} catch (Exception e){
    System.out.println("Caught Exception in main: " + e.getMessage());
}
}
```

Myservice.java

```
package com.example.demo.service;
import org.springframework.stereotype.Service;
@Service
public class MyService {
    public void doSomething() {
        System.out.println("Executing doSomething()...");
    }
    public String getData() {
        System.out.println("Executing getData()...");
        return "Hello from AOP";
    }
    public void throwError() {
        System.out.println("Executing throwError()...");
        throw new RuntimeException("An error occurred!");
    }
}
```

LoggingAspect.java

```
package com.example.demo.aspect;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.*;
import org.springframework.stereotype.Component;

@Aspect
@Component

public class LoggingAspect {

    // POINTCUT

    @Pointcut("execution(* com.example.demo.service.MyService.*(..))")
    public void allMethods() {}

    // BEFORE ADVICE

    @Before("allMethods()")
    public void beforeAdvice() {
        System.out.println("Before Advice: Method is about to execute");
    }

    // AFTER ADVICE

    @After("allMethods()")
    public void afterAdvice() {
        System.out.println("After Advice: Method execution finished");
    }

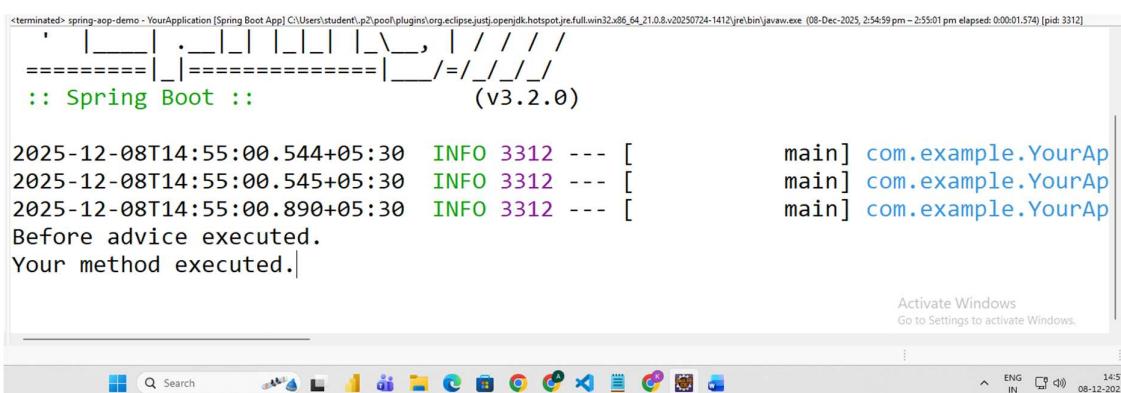
    // AROUND ADVICE

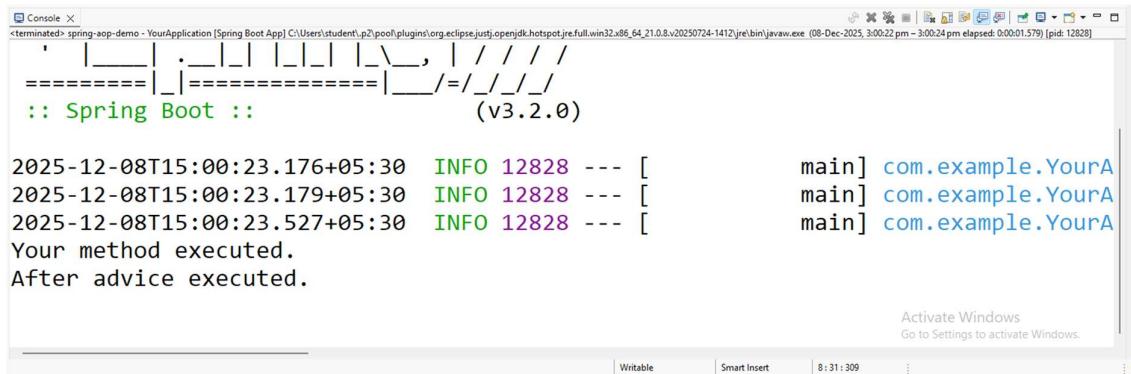
    @Around("execution(* com.example.demo.service.MyService.doSomething(..))")
    public Object aroundAdvice(ProceedingJoinPoint pjp) throws Throwable {
        System.out.println("Around Advice: Before method...");
        Object result = pjp.proceed();
        System.out.println("Around Advice: After method...");
        return result;
    }
}
```

```
    return result;  
}  
  
// AFTER RETURNING  
  
@AfterReturning(  
    pointcut = "execution(* com.example.demo.service.MyService.getData(..))",  
    returning = "value")  
  
public void afterReturningAdvice(Object value) {  
  
    System.out.println("After Returning Advice: Method returned = " + value);  
}  
  
// AFTER THROWING  
  
@AfterThrowing(  
    pointcut = "execution(* com.example.demo.service.MyService.throwError(..))",  
    throwing = "ex")  
  
public void afterThrowingAdvice(Exception ex) {  
  
    System.out.println("After Throwing Advice: Exception = " + ex.getMessage());  
}
```

Output :-

Before output



After output


```

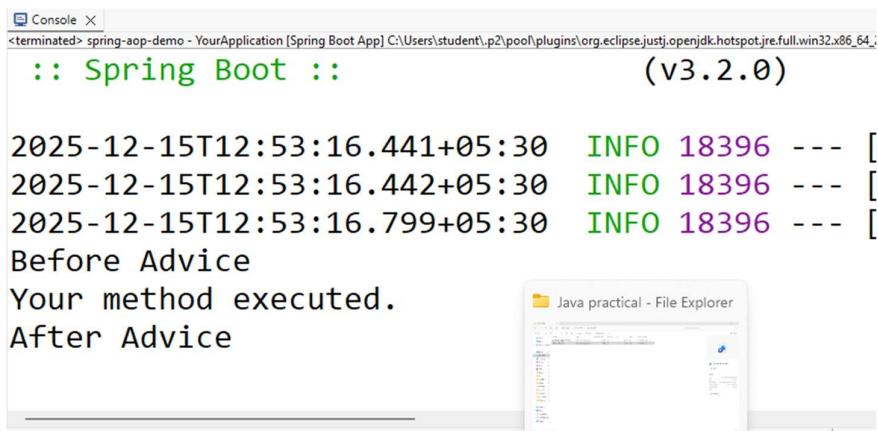
Console X
<terminated> spring-aop-demo - YourApplication [Spring Boot App] C:\Users\student\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (08-Dec-2025, 3:00:22 pm - 3:00:24 pm elapsed: 0:00:01.579) [pid: 12828]
'   |____| .__|_|_|_||_\ \_, | / / / /
=====|_|=====|_|/_=/\_/_/
:: Spring Boot ::          (v3.2.0)

2025-12-08T15:00:23.176+05:30  INFO 12828 --- [           main] com.example.YourA
2025-12-08T15:00:23.179+05:30  INFO 12828 --- [           main] com.example.YourA
2025-12-08T15:00:23.527+05:30  INFO 12828 --- [           main] com.example.YourA
Your method executed.
After advice executed.

Activate Windows
Go to Settings to activate Windows.

Writable Smart Insert 8:31:309 ...

```

Around Advice output


```

Console X
<terminated> spring-aop-demo - YourApplication [Spring Boot App] C:\Users\student\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (08-Dec-2025, 3:00:22 pm - 3:00:24 pm elapsed: 0:00:01.579) [pid: 12828]
:: Spring Boot ::          (v3.2.0)

2025-12-15T12:53:16.441+05:30  INFO 18396 --- [
2025-12-15T12:53:16.442+05:30  INFO 18396 --- [
2025-12-15T12:53:16.799+05:30  INFO 18396 --- [
Before Advice
Your method executed.
After Advice

```

After Returning Advice Output:

```

-----
==> Calling getData() ==
Before Advice: Method is about to execute
Executing getData()...
After Returning Advice: Method returned = Hello from AOP
After Advice: Method execution finished
-----

```

After throwing advice Output:

```

==> Calling throwError() ==
Before Advice: Method is about to execute
Executing throwError()...
After Throwing Advice: Exception = An error occurred!
After Advice: Method execution finished
Caught Exception in main: An error occurred!

```

Pointcuts Output:

```
--- Calling doSomething() ---
Before Advice: Method is about to execute
Executing doSomething()...
After Advice: Method execution finished
-----
--- Calling getData() ===
Before Advice: Method is about to execute
Executing getData()...
After Advice: Method execution finished
-----
--- Calling throwError() ===
Before Advice: Method is about to execute
Executing throwError()...
After Advice: Method execution finished
Caught Exception in main: An error occurred!
```

PRACTICAL NO. 9**Aim :- Assignment based Spring Boot and RESTful Web Services****9.A:-Write a program to create a simple Spring Boot application that prints a message.****Code :-****project1\project1\src\main\java\com\tabish\project1\Project1Application.java**

Project1Application.java

```
package com.tabish.project1;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
@SpringBootApplication  
public class Project1Application {
```

```
    public static void main(String[] args) {  
        SpringApplication.run(Project1Application.class, args);  
    }  
}
```

project1\project1\src\main\java\com\tabish\project1\controller\Testcontroller.java

Testcontroller.java

```
package com.tabish.project1.controller;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
@Controller  
public class Testcontroller {
```

```
    @GetMapping("/greet")  
    public String greet() {  
        return "hello";  
    }  
}
```

project1\project1\src\main\resources\templates\hello.html

Hello.html

```
<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">
<html>
<head>

<title>Mario Theme - Color Choose</title>

<style>

/* 🎨 CHANGE COLORS HERE */

:root {

--bg-color: #4ec0ca; /* sky blue */

--text-color: #703c3c; /* main text color */

--shadow-color: #000000; /* text shadow */

--box-bg: #1fb878; /* question-block yellow */

--box-border: #a13737; /* black border */

}

body {

margin: 0;

padding: 0;

background: var(--bg-color);

font-family: 'Press Start 2P', sans-serif;

text-align: center;

color: var(--text-color);

}

@import

url('https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap');

h1 {

margin-top: 150px;

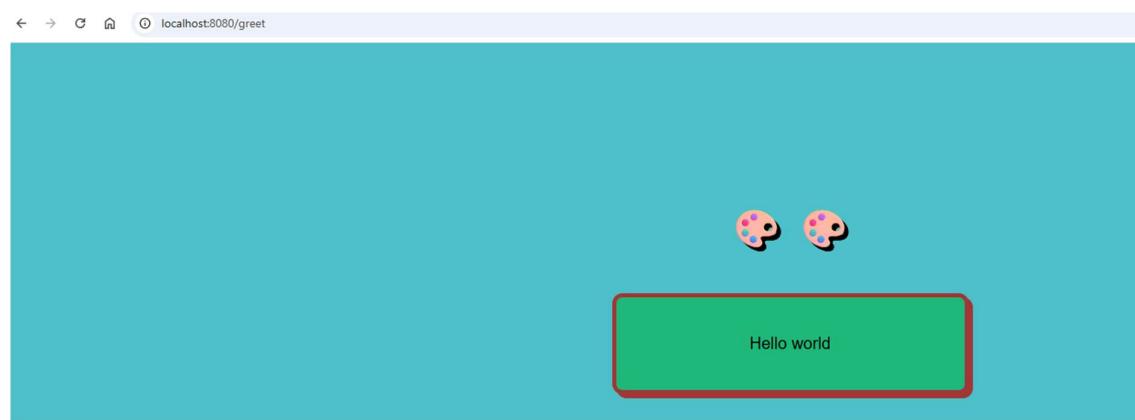
font-size: 40px;

text-shadow: 4px 4px 0 var(--shadow-color);

}
```

```
.mario-box {  
    margin: 40px auto;  
    padding: 20px;  
    width: 300px;  
    background: var(--box-bg);  
    border: 4px solid var(--box-border);  
    border-radius: 10px;  
    box-shadow: 4px 4px 0 var(--box-border);  
    color: black;  
}  
</style>  
</head>  
<body>  
    <h1>👾👾</h1>  
    <div class="mario-box">  
        <p>Hello world</p>  
    </div>  
</body>  
</html>
```

Output :-



AIM:-

9. B:- Write a program to demonstrate RESTful Web Services with spring boot.

C:-Write a program to demonstrate Database Connection with spring boot.

Code :-

project1\src\main\resources\application.properties

```
spring.application.name=project1
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/spring
```

```
spring.datasource.username=root
```

```
spring.datasource.password=gnims
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.show-sql=true
```

project1\src\main\java\com\tabish\project1\repository\studentrepository.java

```
package com.tabish.project1.repository;
```

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

```
import org.springframework.stereotype.Repository;
```

```
import com.tabish.project1.entity.*;;
```

```
@Repository
```

```
public interface studentrepository extends JpaRepository< Student , Integer>{
```

```
}
```

project1\src\main\java\com\tabish\project1\entity\Student.java

```
package com.tabish.project1.entity;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.GeneratedValue;
```

```
import jakarta.persistence.GenerationType;
```

```
import jakarta.persistence.Id;
```

```
@Entity
```

```
public class Student {
```

```
@Id  
@GeneratedValue(strategy=GenerationType.IDENTITY)  
private int id;  
private String name;  
private String city;  
public Student(String city, int id, String name) {  
    this.city = city;  
    this.id = id;  
    this.name = name;  
}  
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 getCity() {  
    return city;  
}  
public void setCity(String city) {  
    this.city = city;  
}
```

```
}
```

project1\src\main\java\com\tabish\project1\controller\Studentcontroller.java

```
package com.tabish.project1.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.tabish.project1.entity.Student;
import com.tabish.project1.service.StudentService;
import java.util.List;

@RestController
@RequestMapping("/students/api")
public class Studentcontroller {

    @Autowired
    private StudentService studentService;

    @PostMapping("/add")
    public Student addStudent(@RequestBody Student student) {
        return studentService.add(student);
    }

    @GetMapping("/all")
    public List<Student> getAllStudents() {
        return studentService.getAll();
    }
}
```

project1\src\main\java\com\tabish\project1\service\StudentService.java

```
package com.tabish.project1.service;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;  
import com.tabish.project1.entity.Student;  
import com.tabish.project1.repository.studentrepository;  
  
@Service  
public class StudentService {  
  
    @Autowired  
    private studentrepository repo;  
  
    public Student add(Student student) {  
  
        return repo.save(student);  
    }  
  
    public List<Student> getAll(){  
  
        return repo.findAll();  
    }  
}
```

project1\pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
https://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
    <modelVersion>4.0.0</modelVersion>  
  
    <parent>  
        <groupId>org.springframework.boot</groupId>  
        <artifactId>spring-boot-starter-parent</artifactId>  
        <version>3.5.7</version>  
        <relativePath/> <!-- lookup parent from repository -->  
    </parent>  
    <groupId>com.tabish</groupId>
```

```
<artifactId>project1</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>project1</name>
<description>Demo project for Spring Boot</description>
<url/>
<licenses>
    <license/>
</licenses>
<developers>
    <developer/>
</developers>
<scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
</scm>
<properties>
    <java.version>17</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    
```

```
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>

<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>

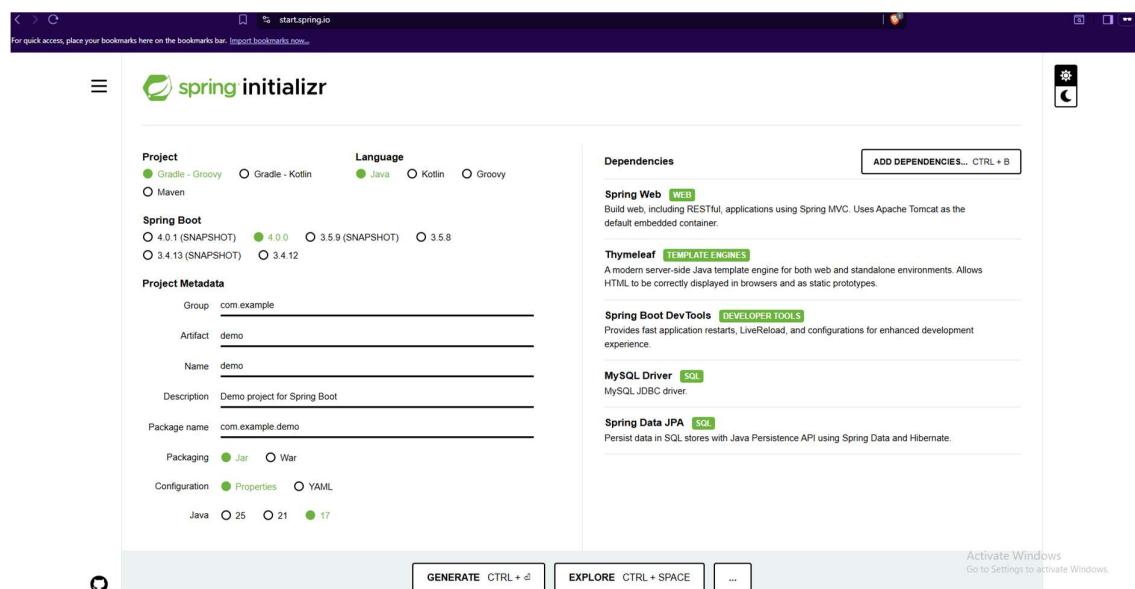
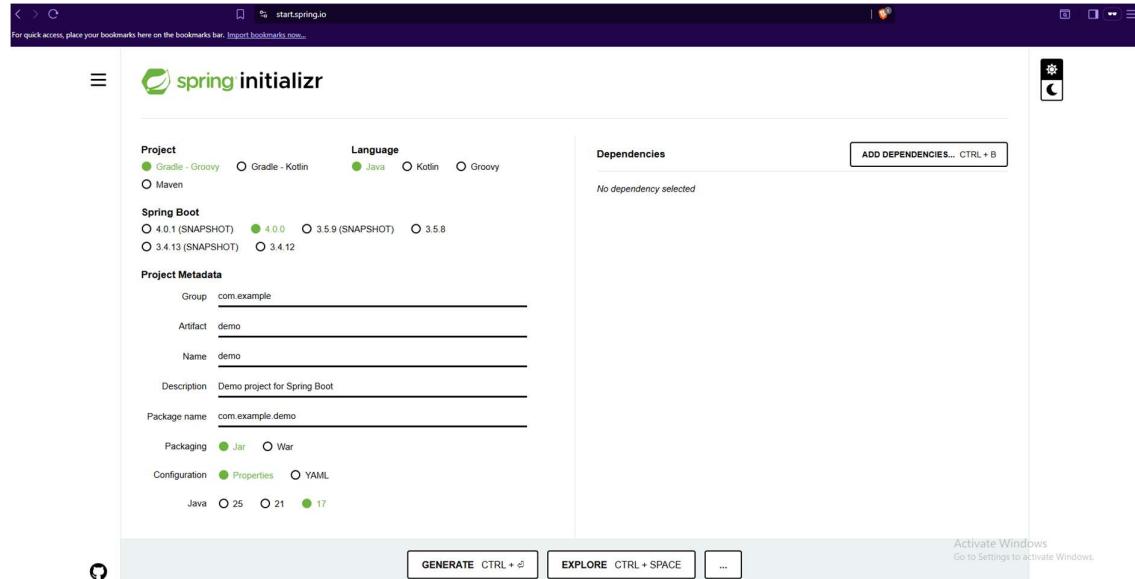
```

```

</plugins>
</build>
</project>

```

Step:-



Output :-

Postman screenshot showing a successful API call to `http://localhost:8080/students/api/add`. The response body is:

```

1 {
2   "id": 1,
3   "name": "tabish",
4   "city": "sion"
5 }
```

MySQL Command Line Client screenshot showing the following session:

```

mysql> create database spring;
Query OK, 1 row affected (0.01 sec)

mysql> use spring;
Database changed

mysql> desc student;
ERROR 1046 (42302): Table 'spring.student' doesn't exist
mysql> select * from student;
+----+-----+-----+
| id | name | city |
+----+-----+-----+
| 1  | tabish | sion |
+----+-----+-----+
1 row in set (0.01 sec)
```