ADVANCED JAVA LAB - MCAL12

INDEX

Program No:	**1**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**FizzBuzz Class**

Date: 12-09-23

Source Code:
FizzBuzzImpl.java:

```
1   class  FizzBuzzModel
2   {
3           //Instance variables
4           private int number;
5
6           //setter and getter
7           public void setNumber(int number)
8           {
9                   this.number=number;
10          }
11
12          public int getNumber()
13          {
14                  return this.number;
15          }
16
17          //Utility Method
18          public String printFizzBuzz()
19          {
20                  String result = ""; //declare an empty string
21                  if(number%3 == 0)
22                  {
23                          result+= "Fizz";
24                  }
25
26                  if(number%5 == 0)
27                  {
28                          result+= "Buzz";
```

```
29                  }
30
31                  //-------------method-1-----------
32                  //if(number%3 == 0 && number%5 == 0)
33                  //{
34                  //      result = "FizzBuzz";
35                  //}
36                  //----------method-2: result+= "Fizz"-----
37
38                  if(result=="")
39                  {
40                          result = ""+number;
41                  }
42                  return result;
43          }
44  }
45
46
47  class FizzBuzzImpl
48  {
49          public static void main(String[] args)
50          {
51                  FizzBuzzModel b1 = new FizzBuzzModel();
52                  for(int i=0;i<=5;i++)
53                  {
54                          b1.setNumber(i);
55                          System.out.println(b1.printFizzBuzz());
56                  }
57
58
59          }
60  }
```

Output:

```
D:\Viraj_1418>javac FizzBuzzImp1.java

D:\Viraj_1418>java FizzBuzzImpl
1
2
Fizz
4
Buzz
```

Program No:	**2**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Employee Class**

Date: 05-09-23

Question: Write a program to generate empid automatically.

Source Code:

EmployeeImpl.java:

```
1   class Employee
2   {
3           //declare a static counter
4           static int counter=100;
5
6           //declare instance variables
7           private int empId;
8           private String empName;
9
10          //decalre a constructor
11          Employee(String empName)
12          {
13                  //auto-increment the empId
14                  counter++;
15
16                  //assign empId
17                  this.empId=counter;
18                  this.empName=empName;
19          }
20
21          //add setters amd getters
22          public void setEmpName(String empName)
23          {
24                  this.empName=empName;
25          }
```

```
26          public String getEmpName()
27          {
28                  return this.empName;
29          }
30          public int getEmpId()
31          {
32                  return this.empId;
33          }
34  }
35
36  class EmployeeImpl
37  {
38          public static void main(String[] args)
39          {
40                  Employee e1=new Employee("Viraj");
41                  Employee e2=new Employee("Ghogale");
42
43                  System.out.println("Employee 1: ");
44                  System.out.println("Emp id:"+e1.getEmpId());
45                  System.out.println("Emp Name:"+e1.getEmpName());
46
47                  System.out.println("\nEmployee 2: ");
48                  System.out.println("Emp id:"+e2.getEmpId());
49                  System.out.println("Emp Name:"+e2.getEmpName());
50
51
52          }
53  }
```

Output:

```
D:\Viraj_1418>javac EmployeeImpl.java

D:\Viraj_1418>java EmployeeImpl
Employee 1:
Emp id:101
Emp Name:Atharva

Employee 2:
Emp id:102
Emp Name:Sawant
```

Program No:	**3**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Character Frequency**

Date: 10-10-23

Source Code:
CharWithFreq.java:

```
1   import java.util.*;
2
3   class CharWithFreq
4   {
5           //method to print chars with frequencies
6           public static void printFreq(String s)
7           {
8                   Map<Character,Integer> m1=new LinkedHashMap<>();
9                   for(int i=0; i<s.length();i++)
10                  {
11                          char c=s.charAt(i);
12                          if(m1.containsKey(c))
13                          {
14                                  int val=m1.get(c);
15                                  val=val+1;
16                                  m1.put(c,val);
17                          }
18                          else
19                          {
20                                  m1.put(c,1);//for the first occurence
21                          }
22                  }
23                          System.out.println(m1);
24          }
25
26          public static void main(String[] args)
27          {
28                  //printFreq("KuchKuch Hota Hai");
```

```
29                  Scanner sc= new Scanner(System.in);
30                  System.out.println("Enter String: ");
31                  String s=sc.nextLine();
32                  printFreq(s);
33
34          }
35  }
36
37
```

Output:

```
D:\Viraj_1418>javac CharWithFreq.java

D:\Viraj_1418>java CharWithFreq
Enter String:
kuch kuch hota hai
{k=2, u=2, c=2, h=4,  =3, o=1, t=1, a=2, i=1}
```

Program No:	**4**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**BankAccount Class**

Date: 05-09-23

Source Code:
BankAccountImpl.java:

```
1   class BankAccount
2   {
3           //----------Instance Variables-------
4           private int accountNumber;
5           private String accountName;
6           private double accountBalance;
7
8           //----------Setters and Getters-------------
9
10          //---------------accountNumber----------------
11          public void setaccountNumber(int accountNumber)
12          {
13                  this.accountNumber = accountNumber;
14          }
15
16          public int getaccountNumber()
17          {
18                  return this.accountNumber;
19          }
20
21          //--------------accountName----------------------
22          public void setaccountName(String accountName)
23          {
24                  this.accountName = accountName;
25          }
26
27          public String getaccountName()
```

```
28          {
29                  return this.accountName;
30          }
31
32          //--------------accountBalance----------------------
33          public void setaccountBalance(double accountBalance)
34          {
35                  this.accountBalance = accountBalance;
36          }
37
38          public double getaccountBalance()
39          {
40                  return this.accountBalance;
41          }
42
43          //--------------Utility Methods------------
44          //---------------deposit-------------
45          public void deposit(double amount)
46          {
47                  this.accountBalance+=amount;
48          }
49          //---------------withdraw-------------
50          boolean withdraw(double amount)
51          {
52
53                  if(this.accountBalance-amount >=0)
54                  {
55                          //withdrawl is possible
56                          //withdraw
57                          this.accountBalance-=amount;
58                          return true;
59                  }
60                  else
61                  {
62                          //withdrawl is not possible
63                          return false;
64                  }
```

```
65          }
66  }
67
68  class BankAccountImpl
69  {
70          public static void main(String[] args)
71          {
72                  BankAccount b1=new BankAccount();
73                  b1.setaccountNumber(101);
74                  b1.setaccountName("Viraj");
75                  b1.setaccountBalance(10000);
76
77                  System.out.println("Ac No: "+b1.getaccountNumber());
78                  System.out.println("Ac Name: "+b1.getaccountName());
79                  System.out.println("Ac Balance: "+b1.getaccountBalance());
80
81                  b1.deposit(5000);
82                  System.out.println("Ac Balance after deposit:
83  "+b1.getaccountBalance());
84                  if(b1.withdraw(1000))
85                  {
86                          System.out.println("Sucessfully withdrwan the amount!!!!");
87                          System.out.println("Updated Balance:
88  "+b1.getaccountBalance());
89                  }
90                  else
91                  {
92                          System.out.println("Insufficient Funds!!!!");
93                  }
          }
  }
```

Output:

```
D:\Viraj_1418>javac BankAccount.java

D:\Viraj_1418>java BankAccountImpl
Ac No: 101
Ac Name: Viraj
Ac Balance: 10000.0
Ac Balance after deposit: 15000.0
Sucessfully withdrwan the amount!!!!
Updated Balance: 14000.0
```

Program No:	**5**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Array Utilities**

Date: 12-09-23

Source Code:
ArrayUtilDemo.java:

```
1   class ArrayUtils
2   {
3           int[] data;
4
5           //set the data
6           public void setData(int[] data)
7           {
8                   this.data=data;
9           }
10
11          //----------------Max Element-----------
12          public int maxEle()
13          {
14                  int max=data[0];
15                  for(int i=1;i<data.length;i++)
16                  {
17                          if(data[i]>max)
18                          {
19                                  max=data[i];
20                          }
21                  }
22                  return max;
23          }
24
25          //-----------------Min Element----------------
26          public int minEle()
27          {
28                  int min=data[0];
```

```
29                  for(int i=1;i<data.length;i++)
30                  {
31                      if(data[i]<min)
32                      {
33                          min=data[i];
34                      }
35                  }
36                  return min;
37          }
38  }
39
40  class ArrayUtilDemo
41  {
42          public static void main(String[] args)
43          {
44                  /*--------------------Default Initialization-----------
45                  int[] arr=new int[3];
46                  System.out.println("Default Initialization");
47
48                  for(int i=0;i<arr.length;i++)
49                  {
50                      System.out.println("Index: "+i+" Value: "+arr[i]);
51                  }
52                  ---------------------------*/
53
54                  int[] arr1={2,4,6};
55                  ArrayUtils a1 = new ArrayUtils();
56                  a1.setData(arr1);
57                  System.out.println("Max of Array: "+a1.maxEle());
58                  System.out.println("Min of Array: "+a1.minEle());
59          }
60  }
```

Output:

```
D:\Viraj_1418>javac ArrayUtilDemo.java

D:\Viraj_1418>java ArrayUtilDemo
Max of Array: 6
Min of Array: 2
```

Program No:	**6**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**MyStack V1**

Date: 26-09-23

Question-1: Create a stack of integers with push and pop functions generic class

Source Code:
MyStackImpl1.java:

```
1   class MyStackV1
2   {
3           int[] data;
4           int size;
5           int tos=-1;
6           MyStackV1(int size)
7           {
8                   this.size=size;
9                   data=new int[size];
10          }
11          void push(int ele)
12          {
13                  if(tos < (size-1))
14                  {
15                          data[++tos] = ele;
16                  }
17                  else
18                  {
19                          System.out.println("Overflow");
20                  }
21          }
22          void pop()
23          {
24                  if(tos >= 0)
25                  {
```

```
26                          System.out.println("Popped: "+data[tos--]);
27                  }
28                  else
29                  {
30                          System.out.println("Underflow");
31
32                  }
33          }
34  }
35
36  class MyStackImpl1
37  {
38          public static void main(String args[])
39          {
40                  MyStackV1 s1 = new MyStackV1(5);
41                  for(int i=0;i<5;i++)
42                  {
43                          s1.push(i*10);
44                  }
45                  s1.pop();
46                  s1.pop();
47                  s1.pop();
48                  s1.pop();
49                  s1.pop();
50                  s1.pop(); //underflow
51          }
52  }
53
54
```

Output:

```
D:\Viraj_1418>javac MyStackImpl1.java

D:\Viraj_1418>java MyStackImpl1
Popped: 40
Popped: 30
Popped: 20
Popped: 10
Popped: 0
Underflow
```

Program No:	**7**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**MyStack V2**

Date: 26-09-23

Question-1: Create a stack of integers with push and pop functions with generic class

Source Code:
MyStackImpl2.java:

```
1   import java.util.*;
2
3   class MyStackV2<T>
4   {
5
6           //a data structure to data
7           ArrayList<T> data;
8           int size;
9           int tos=-1;
10
11          //constructor - to instantiate array with size
12          MyStackV2(int size)
13          {
14                  this.size=size;
15                  data=new ArrayList<T>(size);
16          }
17
18          //push
19          void push(T ele)
20          {
```

```
21                  //increment tos
22                  tos++;
23                  if(tos<size)
24                  {
25                          //push the element
26                          data.add(tos,ele);
27                  }
28                  else
29                  {
30                          System.out.println("Overflow");
31                  }
32          }//end of push
33
34          //get and return the popped value
35          T pop()
36          {
37                  tos--;
38                  if(tos>=0)
39                  {
40                          return data.remove(tos);
41                  }
42                  else
43                  {
44                          System.out.println("Underflow");
45                          return null;
46                  }
47          }//end of pop
48
49  }
50
51  class MyStackImpl2
52  {
53          public static void main(String[] args)
54          {
55                  MyStackV2<Integer> s1 = new MyStackV2<>(5);
56
57                  for(int i=0;i<=5;i++)
```

```
58                  {
59                          s1.push(i*10);
60                  }
61
62                  for(int j=0;j<=5;j++)
63                  {
64                          System.out.println("Popped: "+s1.pop());
65                  }
66          }
67
68  }
69
70
```

Output:

```
D:\Viraj_1418>javac MyStackImpl2.java

D:\Viraj_1418>java MyStackImpl2
Overflow
Popped: 40
Popped: 30
Popped: 20
Popped: 10
Popped: 0
Underflow
Popped: null
```

Program No:	**8**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**LinkedList Class**

Date:

Source Code:
LinkedListTest.java

```
1  class Node<T>
2  {
3          // data
4          T data;
5          // address of next node
6          Node<T> next;
7          //Constructor
8          Node(T data)
9          {
10                 this.data=data;
11                 next=null;
12         }
13 }
14 class MyLinkedList<T>
15 {
16         // head attribute
17         Node<T> head;
18         int size;
19         int getsize()
20         {
21                 return size;
22         }
23         MyLinkedList()
24         {
25                 size=0;
26                 head=null;
27         }
28         //add a node at the end
29         void add(T data)
30         {
```

```
31                  //create a temporary node
32                  Node<T> temp=new Node<>(data);
33                  //check is it the first node
34                  if(head==null)
35                  {
36                          head=temp;
37                          System.out.println("first Node added");
38                  }
39                  else
40                  {
41                          //not a first node - reach to the end
42                          Node<T> x=head;
43                          while(x.next!=null)
44                          {
45                                  x=x.next;
46                          }
47                          x.next=temp;
48                          System.out.println("Next Node Added");
49
50                  }
51                  size++;
52          }
53          public String toString()
54          {
55                  String result="[";
56                  Node<T> x=head;
57                  if(head==null)
58                  {
59                          result=result+"]";
60                          return result;
61                  }
62                  while(x.next!=null)
63                  {
64                          result=result+x.data+"->";
65                          x=x.next;
66                  }
67                  result+=x.data+"]";
68                  return result;
69          }
70  }
71  class LinkedListTest
72  {
73          public static void main(String[] args)
```

```
74          {
75                  MyLinkedList<Integer> I1=new MyLinkedList<>();
76                  I1.add(10);
77                  I1.add(20);
78                  I1.add(30);
79                  System.out.println("Size: "+I1.getsize());
80                  System.out.println("List: "+I1.toString());
81          }
82  }
```

Output:

```
D:\Viraj_1418>javac LinkedListTest.java

D:\Viraj_1418>java LinkedListTest
first Node added
Next Node Added
Next Node Added
Size: 3
List: [10->20->30]
```

Program No:	**9**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Collections Demo**

Date:

Source Code:
LinkedListDemo.java

```
1  import java.util.*;
2  class LinkedListDemo
3  {
4          public static void main(String[] args)
5          {
6                  List<String> names=new LinkedList<>();
7                  Scanner sc=new Scanner(System.in);
8                  for(int i=0;i<5;i++)
9                  {
10                         System.out.println("Enter name: ");
11                         String n = sc.next();
12                         names.add(n);
13                 }
14                 System.out.println("\nBefore Sorting: "+names+"\n");
15
16                 Iterator i1=names.iterator();
17                 while(i1.hasNext())
18                 {
19                         System.out.println(i1.next());
20                 }
21                 Collections.sort(names);
22                 System.out.println("\nAfter Sorting in Ascending Order: "+names);
23
24                 Collections.sort(names,Collections.reverseOrder());
25                 System.out.println("\nAfter Sorting in Descending Order:
26 "+names);
27         }
   }
```

Output:

```
D:\Viraj_1418>javac  LinkedListDemo.java

D:\Viraj_1418>java  LinkedListDemo
Enter name:
viraj
Enter name:
snehee
Enter name:
prathamesh
Enter name:
chaitanya
Enter name:
yukta

Before Sorting: [viraj, snehee, prathamesh, chaitanya, yukta]

viraj
snehee
prathamesh
chaitanya
yukta

After Sorting in Ascending Order: [chaitanya, prathamesh, snehee, viraj, yukta]

After Sorting in Descending Order: [yukta, viraj, snehee, prathamesh, chaitanya]
```

Program No:	**10**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Calculator Using Lambda**

Date: 03-10-23

Source Code:
CalculatorImpl.java:

```
1  interface Calculator
2  {
3          double op(double A , double b);
4  }
5
6  class CalculatorImpl
7  {
8          public static void main(String[] args)
9          {
10                 //create an instance of calcultor interface
11                 Calculator addImpl = (a,b)-> a+b;
12                 Calculator subImpl = (a,b)-> a-b;
13                 Calculator mulImpl = (a,b)-> {return a*b;};
14
15                 //invoke the method
16                 System.out.println("Addition: "+addImpl.op(10,20));
17                 System.out.println("Subtraction: "+subImpl.op(10,20));
18                 System.out.println("Multiplication: "+mulImpl.op(10,20));
19         }
20 }
21
22
```

Output:

```
D:\Viraj_1418>javac CalculatorImpl.java

D:\Viraj_1418>java CalculatorImpl
Addition: 30.0
Subtraction: -10.0
Multiplication: 200.0
```

Program No:	**11**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Pair Class**

Date: 03-10-23

Source Code:
PairImpl.java:

```
1   class Pair<K,V>
2   {
3           //variables
4           K key;
5           V value;
6
7           void setKey(K key)
8           {
9                   this.key=key;
10          }
11
12          K getKey()
13          {
14                  return this.key;
15          }
16
17          void setValue(V value)
18          {
19                  this.value=value;
20          }
21
22          V getValue()
23          {
24                  return this.value;
25          }
26
27          void print()
28          {
```

```
29                  System.out.println("Key: "+key+"\nValue: "+value);
30          }
31  }
32
33
34  class PairImpl
35  {
36          public static void main(String[] args)
37          {
38                  Pair <Integer,String> p1 = new Pair<>();
39                  p1.setKey(1418);
40                  p1.setValue("Viraj");
41                  p1.print();
42          }
43  }
44
45
46
47
48
49
50
51
52
53
54
```

Output:

```
D:\Viraj_1418>javac PairImpl.java

D:\Viraj_1418>java PairImpl
Key: 1418
Value: Viraj
```

Program No:	**12**
Roll No :	**1418**
Title of Program :	**Collections & Generics**
Objective :	**Box**

Date: 03-10-23
Source Code:

BoxImpl.java:

```
1   //Design a generic box class with one generic type T. Implementaion will be
2   String,Integer type(5 marks)
3
4   class Box<T>
5   {
6           T item;
7
8           //setter and getter
9           void setItem(T item)
10          {
11                  this.item=item;
12          }
13
14          T getItem()
15          {
16                  return this.item;
17          }
18
19  }
20
21  class BoxImpl
22  {
23          public static void main(String[] args)
24          {
25                  Box<String> b1 = new Box<>();
26                  Box<Integer> b2 = new Box<>();
27                  b1.setItem("Viraj Ghogale");
```

```
28                  b2.setItem(1418);
29                  System.out.println("Box of String: "+b1.getItem());
30                  System.out.println("Box of Integer: "+b2.getItem());
31
32          }
    }
```

Output:

```
D:\Viraj_1418>javac BoxImpl.java

D:\Viraj_1418>javac BoxImpl.java

D:\Viraj_1418>java BoxImpl
Box of String: Viraj Ghogale
Box of Integer: 1418
```

Program No:	**13**
Roll No :	**1418**
Title of Program :	**JSP**
Objective :	**Multiplication Table**

Date:

Source Code:

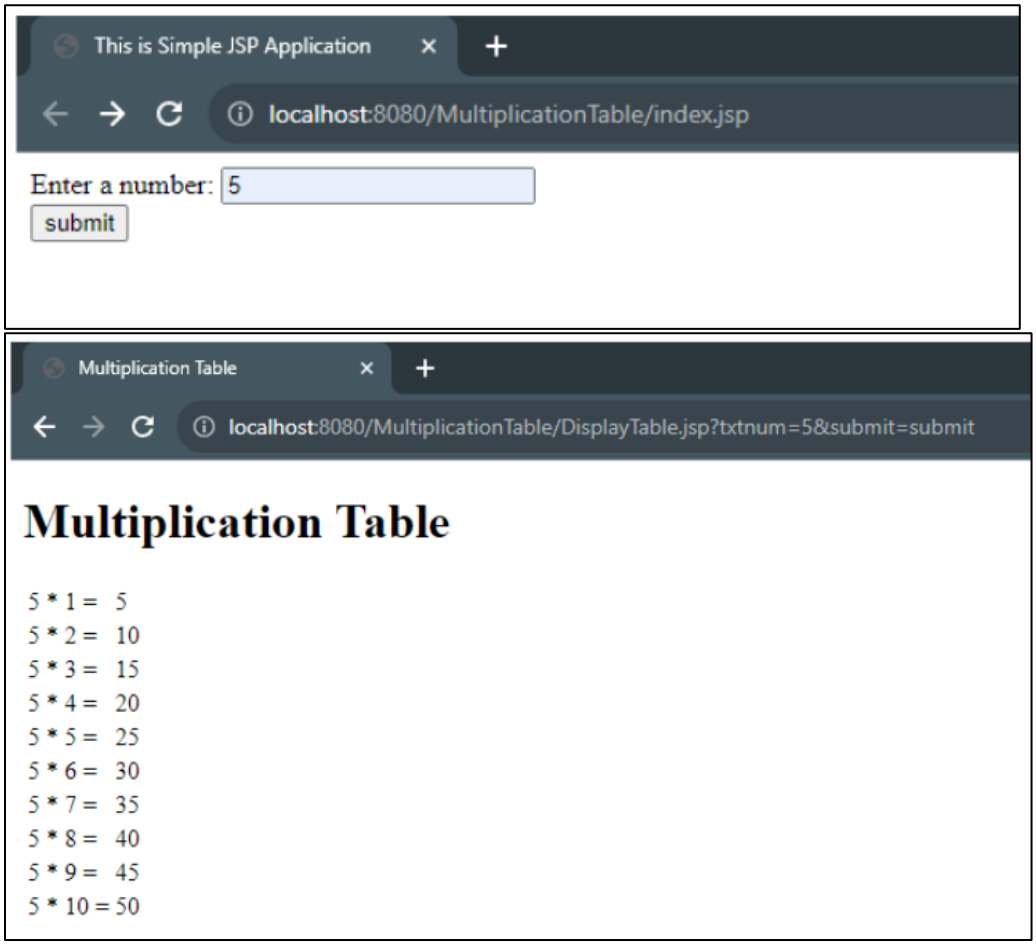
Index.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>This is Simple JSP Application </title>
8  </head>
9
10 <body>
11 <form method="get" action="DisplayTable.jsp">
12 Enter a number: <input type="text" name="txtnum"><br>
13 <input type="submit" name="submit" value="submit">
14 </form>
15 </body>
16 </html>
17
```

DisplayTable.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Multiplication Table</title>
8  </head>
9  <body>
10
11    <%!
12       //declare the variables
13       int num;
14    %>
15
16    <!-- Fetch the value from the text box -->
17    <Table>
18    <%
19       num = Integer.parseInt(request.getParameter("txtnum"));
20       out.println("<h1>Multiplication Table</h1>");
21       for(int i=1;i<=10;i++)
22       {
23    %>
24       <tr>
25       <td><%= num %> * <%=i %> = </td>
26       <td><%= (num*i) %></td>
27       </tr>
28    <%
29       }
30    %>
31    </Table>
32 </body>
33 </html>
34
```

Output:


This is Simple JSP Application
localhost:8080/MultiplicationTable/index.jsp
Enter a number: 5
submit
Multiplication Table
localhost:8080/MultiplicationTable/DisplayTable.jsp?txtnum=5&submit=submit
Multiplication Table
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

Program No:	**14**
Roll No :	**1418**
Title of Program :	**JSP**
Objective :	**Number Guesser**

Date:

Source Code:

Index.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Number Guesser Game</title>
8  </head>
9  <body>
10
11 <form action="NextGuess.jsp">
12 <h1>Number Guesser</h1><br>
13 <h2>Select a number between 0 and 100</h2><br>
14
15 <input type="submit" name="btnSub" value="Start">
16 <input type="hidden" name="low" value="0">
17 <input type="hidden" name="high" value="100">
18 <input type="hidden" name="cntr" value="1">
19
20 </form>
21 </body>
22 </html>
```

NextGuess.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Next Guess</title>
8  </head>
9  <body>
10 <%! int guess; %>
11 <%
12    //fetch all the form fields
13    int hi = Integer.parseInt(request.getParameter("high"));
14    int lo = Integer.parseInt(request.getParameter("low"));
15    int cnt = Integer.parseInt(request.getParameter("cntr"));
16
17    guess =  (hi+lo)/2;
18
19 %>
20
21 <form action="CheckResult.jsp">
22 My Guess no: <%= cnt++ %> is: <%= guess %> <br>
23 <h2>How is my Guess?</h2><br>
24
25 <input type="radio" name="guesser" value="too_low" onclick="submit()">Too
26 Low
27 <input type="radio" name="guesser" value="too_high"
28 onclick="submit()">Too High
29 <input type="radio" name="guesser" value="perfect"
30 onclick="submit()">Perfect
31 <input type="hidden" name="low" value="<%= lo %>">
32 <input type="hidden" name="high" value="<%= hi %>">
33 <input type="hidden" name="cntr" value="<%= cnt %>">
34 </form>
35 </body>
36 </html>
```

CheckResult.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Check Result</title>
8  </head>
9  <body>
10 <%
11    int hi = Integer.parseInt(request.getParameter("high"));
12    int lo = Integer.parseInt(request.getParameter("low"));
13    int cnt = Integer.parseInt(request.getParameter("cntr"));
14    int guess = (hi+lo)/2;
15    String check = request.getParameter("guesser");
16
17    if(check.equals("perfect"))
18    {
19 %>
20    <jsp:forward page="Winner.jsp">
21       <jsp:param value="<%= cnt %>" name="cntr"/>
22    </jsp:forward>
23 <%
24    }
25    else if(hi<=lo || lo>=hi)
26    {
27 %>
28    <jsp:forward page="Cheater.jsp" />
29
30 <%
31    }
32    else
33    {
34       if(check.equals("too_low"))
35       {
36          lo=guess;
37       }
38       else
39       {
40          hi=guess;
41       }
```

```
42  %>
43     <jsp:forward page="NextGuess.jsp">
44        <jsp:param value="<%= hi %>" name="high"/>
45        <jsp:param value="<%= lo %>" name="low"/>
46        <jsp:param value="<%= cnt %>" name="cntr"/>
47     </jsp:forward>
48
49  <%
50     }
51  %>
52
53  </body>
54  </html>
```

Winner.jsp

```
1   <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2    pageEncoding="ISO-8859-1"%>
3   <!DOCTYPE html>
4   <html>
5   <head>
6   <meta charset="ISO-8859-1">
7   <title>Winner</title>
8   </head>
9   <body>
10
11  <h1>I guessed in <%= request.getParameter("cntr") %> attempts </h1>
12  <a href="Index.jsp">Play again</a>
13  </body>
14  </html>
```

Cheater.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Cheater</title>
8  </head>
9  <body>
10  <h2>You are a cheater! Stay Away</h2>
11 </body>
12 </html>
```

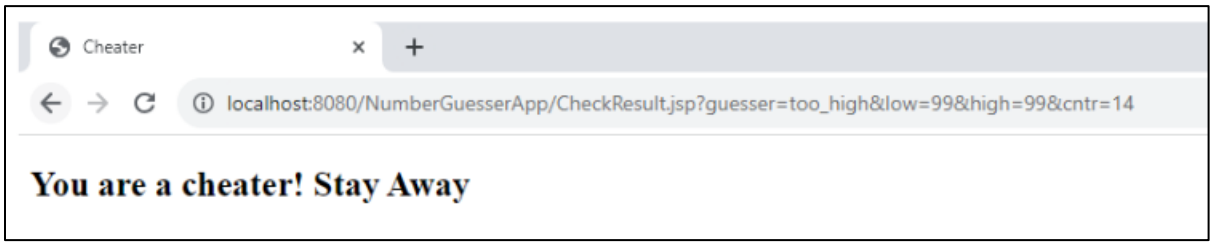
Output:


Number Guesser Game
localhost:8080/NumberGuesserApp/Index.jsp
Number Guesser
Select a number between 0 and 100
Start


Next Guess
localhost:8080/NumberGuesserApp/NextGuess.jsp?btnSub=Start&low=0&high=100&cntr=1
My Guess no: 1 is: 50
How is my Guess?
Too Low
Too High
Perfect


Next Guess
localhost:8080/NumberGuesserApp/CheckResult.jsp?guesser=too_low&low=0&high=100&cntr=2
My Guess no: 2 is: 75
How is my Guess?
Too Low
Too High
Perfect


Winner
localhost:8080/NumberGuesserApp/CheckResult.jsp?guesser=perfect&low=75&high=100&cntr=4
I guessed in 4 attempts
Play again


Cheater
localhost:8080/NumberGuesserApp/CheckResult.jsp?guesser=too_high&low=99&high=99&cntr=14
You are a cheater! Stay Away

Program No:	**15**
Roll No :	**1418**
Title of Program :	**JSP**
Objective :	**Counter using Cookies**

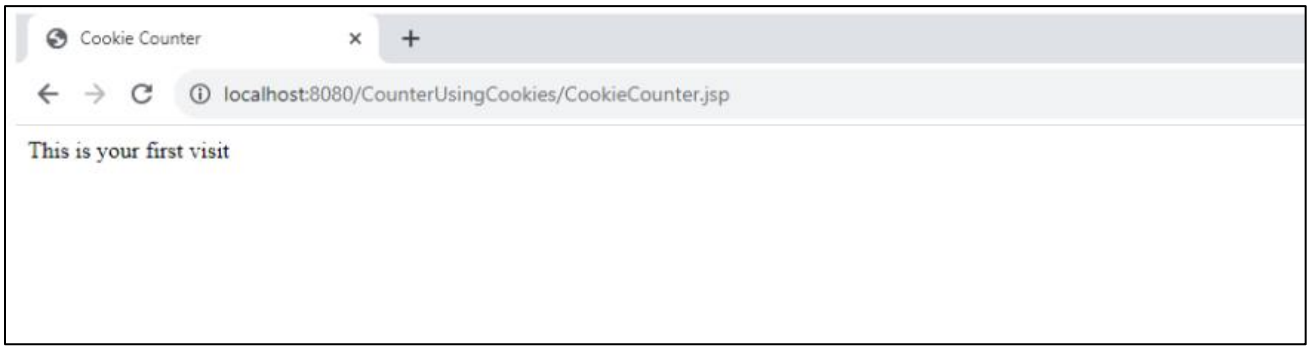
Date:

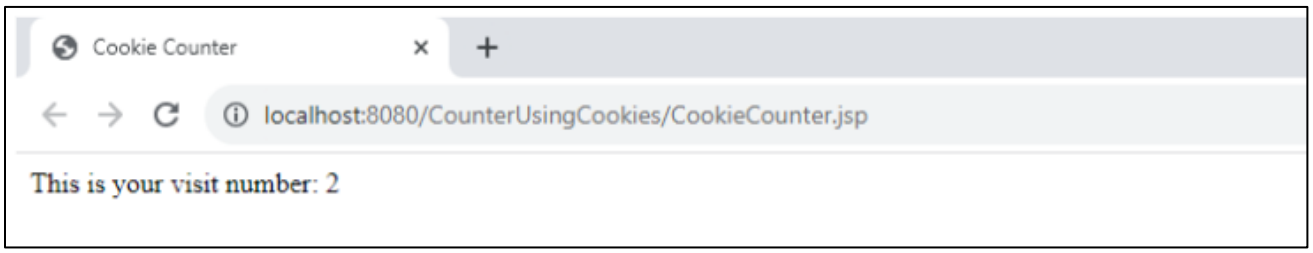
Source Code:
CookieCounter.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1" import ="javax.servlet.http.*" %>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Cookie Counter</title>
8  </head>
9  <body>
10 <%
11 Cookie[]  cookies = request.getCookies();
12  //check whether it is the first request
13
14 if(cookies == null)
15 {
16       //this is your request
17       out.println("This is your first visit");
18
19       //create a cookie
20
21       Cookie c1=new Cookie("cntr","1");
22
23       //add the cookie to the response
24       response.addCookie(c1);
25 }
26 else
27    {
28       for(Cookie temp:cookies)
29       {
30          if(temp.getName().equals("cntr"))
31          {
32             int val = Integer.parseInt(temp.getValue());
```

```
33              val++;
34              out.println("This is your visit number: "+val);
35              temp.setValue(""+val);//convert value to string
36              response.addCookie(temp);
37              break;
38          }
39      }
40  }
41
42  %>
43  </body>
44  </html>
45
46
47
```

Output:


Cookie Counter
localhost:8080/CounterUsingCookies/CookieCounter.jsp
This is your first visit


Cookie Counter
localhost:8080/CounterUsingCookies/CookieCounter.jsp
This is your visit number: 2

Program No:	**16**
Roll No :	**1418**
Title of Program :	**JSP**
Objective :	**Login Application with Beans**

Source Code:
LoginBean.java

```
1  package edu.met;
2  public class LoginBean {
3   private String uname;
4   private String password;
5   public LoginBean() {
6   super();
7   }
8   public String getUname() {
9   return uname;
10  }
11  public void setUname(String uname) {
12  this.uname = uname;
13  }
14  public String getPassword() {
15  return password;
16  }
17  public void setPassword(String password) {
18  this.password = password;
19  }
20  public boolean validate() {
21  if ("atharva".equals(this.uname) && "password".equals(this.password)) {
22  return true;
23  } else {
24  return false;
25  }
26  }
27 }
```

LoginPage.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6   <title>Login Page</title>
7  </head>
8  <body>
9   <form action="Controller.jsp" method="post">
10  <h1>MET ICS</h1>
11  User Name: <input type="text" name="txtUname">
12  <br><br>
13  Password: <input type="password" name="txtPwd">
14  <br><br>
15  <input type="submit" name="btnSubmit" value="Login">
16  </form>
17 </body>
18 </html>
```

Controller.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <%@ page import="edu.met.LoginBean" %>
4  <!DOCTYPE html>
5  <html>
6  <body>
7   <jsp:useBean id="userB" class="edu.met.LoginBean" />
8   <jsp:setProperty property="*" name="userB" />
9   <%
10  if (userB.validate()) {
11  response.sendRedirect("Welcome.jsp");
12  } else {
13  response.sendRedirect("Invalid.jsp");
14  }
15  %>
16 </body>
17 </html>
```

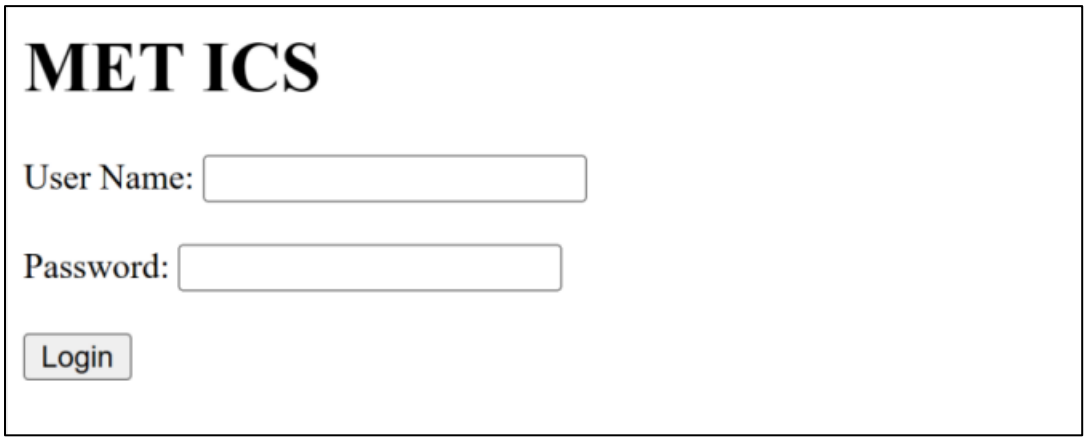
Welcome.jsp

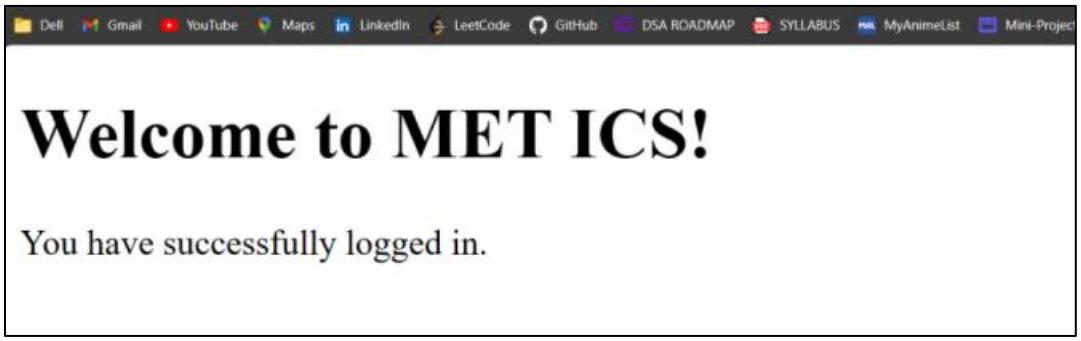
```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <body>
6   <h1>Welcome to MET ICS!</h1>
7   <p>You have successfully logged in.</p>
8  </body>
9  </html>
```

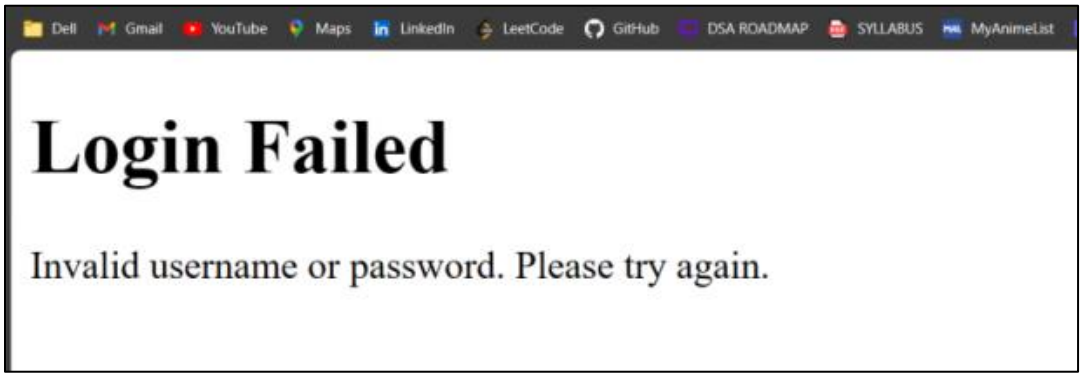
Invalid.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <body>
6   <h1>Login Failed</h1>
7   <p>Invalid username or password. Please try again.</p>
8  </body>
9  </html>
```

Output:


MET ICS
User Name:
Password:
Login


Dell
Gmail
YouTube
Maps
LinkedIn
LeetCode
GitHub
DSA ROADMAP
SYLLABUS
MyAnimeList
Mini-Project
Welcome to MET ICS!
You have successfully logged in.


Dell
Gmail
YouTube
Maps
LinkedIn
LeetCode
GitHub
DSA ROADMAP
SYLLABUS
MyAnimeList
Login Failed
Invalid username or password. Please try again.

Program No:	**17**
Roll No :	**1418**
Title of Program :	**JSP**
Objective :	**Login Application**

Date:

Source Code:
Index.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title> Home Page </title>
8  </head>
9  <body>
10
11 <form method="get" action="CheckLogin.jsp">
12 <h1>MET ICS </h1>
13 <br>
14 Username: <input type="text" name="txtuname">
15 <br>
16 <br>
17 Password: <input type ="password" name="txtpassword">
18 <br>
19 <br>
20 <input type="submit" name="btnSubmit" value="Login">
21 </form>
22
23 </body>
24 </html>
```

CheckLogin.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2  pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Check Login Page</title>
8  </head>
9  <body>
10
11 <%! String uname,pwd; %>
12
13 <%
14       uname = request.getParameter("txtuname");
15       pwd = request.getParameter("txtpassword");
16
17       if(uname.equals("Viraj")&& pwd.equals("1234"))
18       {
19 response.sendRedirect("Welcome.jsp?txtuname="+uname);
20       }
21       else
22       {
23          response.sendRedirect("Invalid.jsp");
24       }
25 %>
26 </body>
27 </html>
28
```

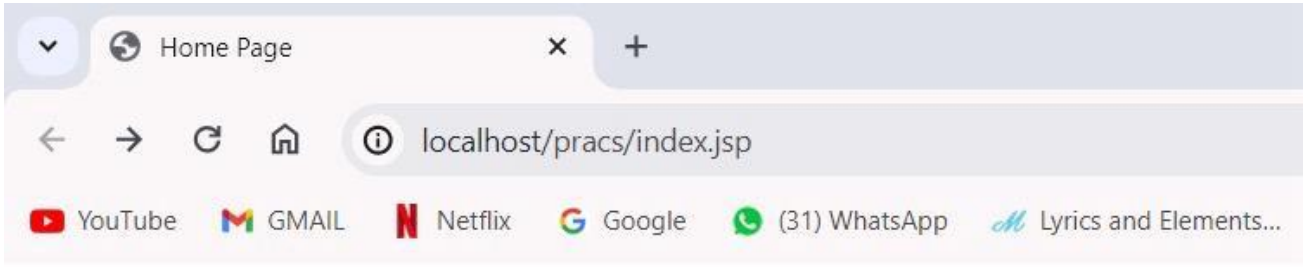
Welcome.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Welcome Page</title>
8  </head>
9  <body>
10 <%
11 String name=request.getParameter("txtuname");
12 %>
13 <h1> Welcome:<%= name %></h1>
14
15 </body>
16 </html>
17
18
```

Invalid.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-
2  1" pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Invalid Page</title>
8  </head>
9  <body>
10
11 <h1> Invalid credentials:(</h1> <br> <a href="Index.jsp">Go to Home
12 Page!</a>
13 </body>
14 </html>
```

Output:

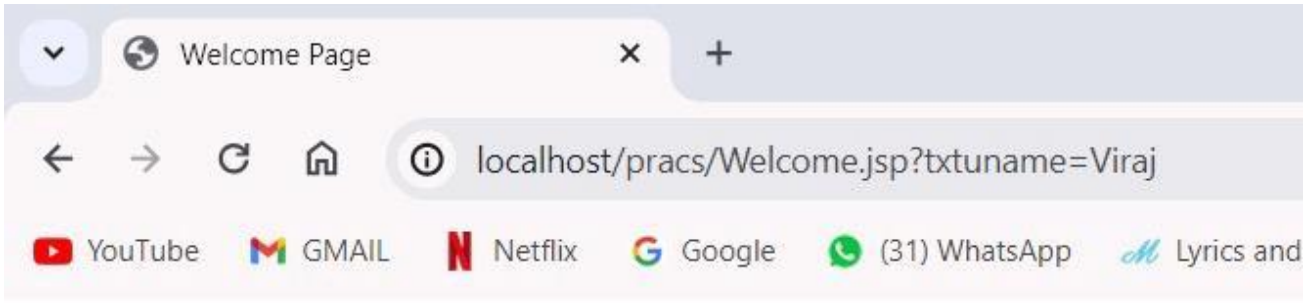


MET ICS

Username: Viraj

Password: ••••

Login



Welcome:Viraj

Program No:	**18**
Roll No :	**1418**
Title of Program :	**JSP**
Objective :	**Database Connectivity using JSTL**

Date:

Source Code:

Index.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <body>
6  <h1>Student Database</h1>
7  <a href="ShowStudents.jsp">
8  <button>Display Student Record</button>
9  </a>
10 <br>
11 <br>
12 <a href="InsertRecord.jsp">
13 <button>Insert New Record</button>
14 </a>
15 </body>
16 </html>
```

ShowStudents.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3  <%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
4  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
5  <!DOCTYPE html>
6  <html>
7  <body>
8  <sql:setDataSource var="myDs" driver="org.postgresql.Driver"
9  url="jdbc:postgresql://localhost:5432/postgres" user="postgres"
10 password="password"
11 scope="application"></sql:setDataSource>
```

```
12 <sql:query var="rs" dataSource="${myDs}">SELECT * FROM
13 students</sql:query>
14 <table border="1">
15 <tr>
16 <th>ID</th>
17 <th>Name</th>
18 <th>City</th>
19 <th>Age</th>
20 </tr>
21 <c:forEach var="row" items="${rs.rows}">
22 <tr>
23 <td><c:out value="${row.sid}"/></td>
24 <td><c:out value="${row.sname}"/></td>
25 <td><c:out value="${row.scity}"/></td>
26 <td><c:out value="${row.sage}"/></td>
27 </tr>
28 </c:forEach>
29 </table>
30 <br>
31 <br>
32 <a href="index.jsp">
33 <button>Back</button>
34 </a>
35 </body>
36 </html>
```

InsertRecord.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <body>
6  <h1>Enter Student Details</h1>
7  <form action="AddRecord.jsp" method="post">
8  ID: <input type="text" name="txtId"/><br><br>
9  Name: <input type="text" name="txtName"/><br><br>
10 City: <input type="text" name="txtCity"/><br><br>
11 Age: <input type="text" name="txtAge"/><br><br>
12 <input type="submit" name="sub" value="Add a Record">
13 </form>
```

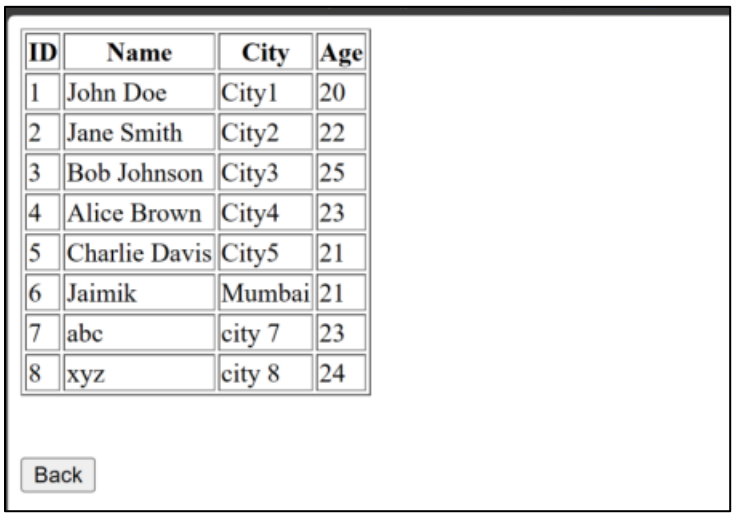
```
14 <br>
15 <a href="index.jsp">
16 <button>Back</button>
17 </a>
18 </body>
19 </html>
```

AddRecord.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
4  <%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
5  <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
6  <!DOCTYPE html>
7  <html>
8  <body>
9  <sql:setDataSource
10 var="myDs"
11 driver="org.postgresql.Driver"
12 url="jdbc:postgresql://localhost:5432/postgres"
13 user="postgres"
14 password="password"
15 scope="application"/>
16 <fmt:parseNumber var="id" value="${param.txtId}"/>
17 <fmt:parseNumber var="age" value="${param.txtAge}"/>
18 <sql:update var="rowCnt" dataSource="${myDs}">
19 INSERT INTO students VALUES(?,?,?,?);
20 <sql:param value="${id}"/>
21 <sql:param value="${param.txtName}"/>
22 <sql:param value="${param.txtCity}"/>
23 <sql:param value="${age}"/>
24 </sql:update>
25 <c:if test="${rowCnt>0}">
26 <c:out value="Inserted"/>
27 </c:if>
28 <br>
29 <br>
30 <a href="index.jsp">
31 <button>Back</button>
32 </a>
```

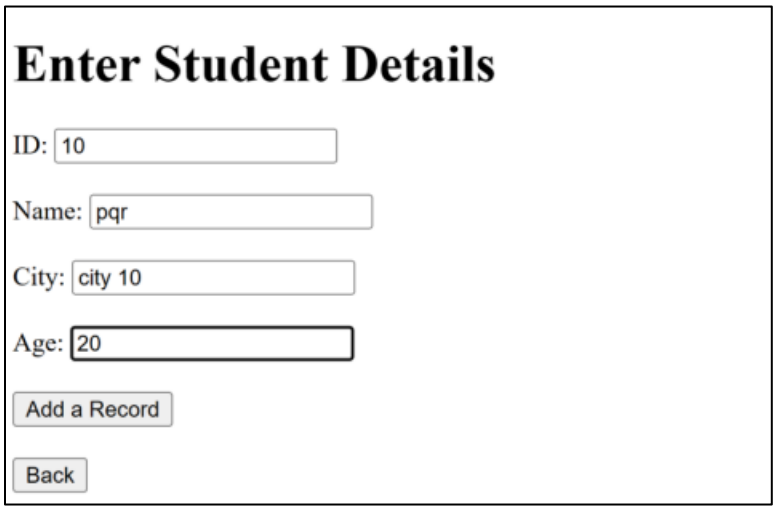
```
33 </body>
34 </html>
```

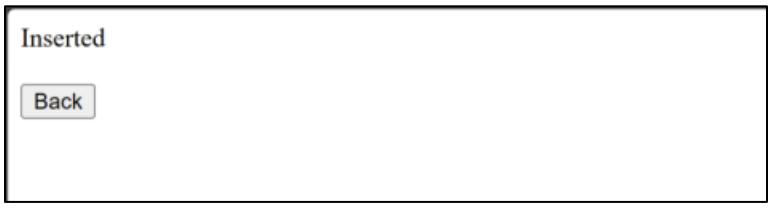
Output:



ID	Name	City	Age
1	John Doe	City1	20
2	Jane Smith	City2	22
3	Bob Johnson	City3	25
4	Alice Brown	City4	23
5	Charlie Davis	City5	21
6	Jaimik	Mumbai	21
7	abc	city 7	23
8	xyz	city 8	24

Back


Enter Student Details
ID: 10
Name: pqr
City: city 10
Age: 20
Add a Record
Back


Inserted
Back

Program No:	**19**
Roll No :	**1418**
Title of Program :	**Spring**
Objective :	**Mobile Application**

Date:

Source Code:
Mobile.java

```
1  package edu.met;
2
3  public class Mobile {
4
5     String brand;
6     String model;
7
8     //Parameterized Constructor using fields
9     public Mobile(String brand, String model) {
10       super();
11       this.brand = brand;
12       this.model = model;
13    }
14
15    //toString method
16    @Override
17    public String toString() {
18       return "Mobile [brand=" + brand + ", model=" + model + "]";
19    }
20 }
```

MobileStore.java

```
1  package edu.met;
2
3  public class MobileStore {
4      int id;
5      Mobile m1;
6      double price;
7
8      //Parameterized Constructor using fields
9      public MobileStore(int id, Mobile m1, double price) {
10         super();
11         this.id = id;
12         this.m1 = m1;
13         this.price = price;
14     }
15
16     //toString method
17     @Override
18     public String toString() {
19         return "MobileStore [id=" + id + ", m1=" + m1 + ", price=" + price + "]";
20     }
21
22 }
```

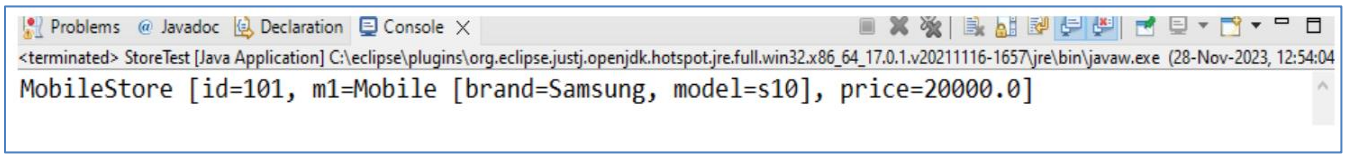
AppCtx.XML

```
?xml version="1.0" encoding="UTF-8"?>
<beans  xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">


<bean id="mobile" class="edu.met.Mobile">
<constructor-arg index="0" value="Samsung"></constructor-arg>
<constructor-arg value="s10"></constructor-arg>

</bean>

<bean id="mobileS" class="edu.met.MobileStore">
<constructor-arg value="101">
</constructor-arg>

<constructor-arg>
<ref bean="mobile"/>
</constructor-arg>

<constructor-arg value="20000">
</constructor-arg>
</bean>
</beans>
```

StoreTest.java

```
1  package edu.met;
2  import org.springframework.context.*;
3  import org.springframework.context.support.*;
4
5  public class StoreTest {
6     private static ApplicationContext ctx;
7     public static void main(String[] args) {
8        // TODO Auto-generated method stub
9        ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
10       MobileStore m1 = (MobileStore)ctx.getBean("mobileS");
11       System.out.println(m1);
12    }
13 }
```

Output:



Program No:	**20**
Roll No :	**1418**
Title of Program :	**Spring**
Objective :	**Singer Application**

Date:

Source Code:
Singer.java

```
1  package edu.met;
2
3  public class Singer {
4     int id;
5     String name;
6
7     //Getters and Setters
8     public int getId() {
9        return id;
10    }
11    public void setId(int id) {
12       this.id = id;
13    }
14    public String getName() {
15       return name;
16    }
17    public void setName(String name) {
18       this.name = name;
19    }
20
21    //Constructor
22    public Singer(int id, String name) {
23       super();
24       this.id = id;
25       this.name = name;
26    }
27    public Singer() {
28       super();
29    @Override
30    public String toString() {
31       return "Singer{name=" + getName() + ", id=" + getId() + "}";
```

```
32  }
33  }
34
35
36
37
```

Appctx.xml

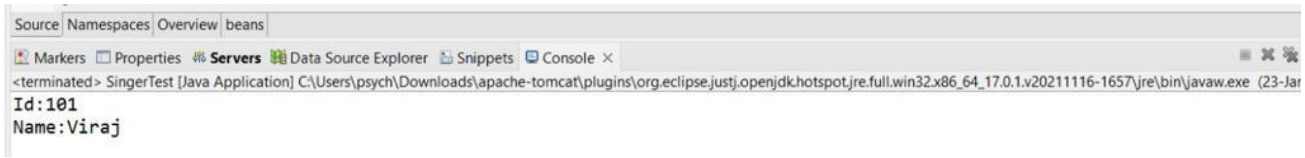
```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <beans
3      xmlns="http://www.springframework.org/schema/beans"
4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      xmlns:p="http://www.springframework.org/schema/p"
6      xsi:schemaLocation="http://www.springframework.org/schema/beans
7              http://www.springframework.org/schema/beans/spring-beans-
8   3.0.xsd">
9   <bean id="SingerBean" class="edu.met.Singer" scope="prototype">
10  <property name="id" value="101"></property>
11  <property name="name" value="Viraj "></property>
12  </bean>
13  </beans>
14
```

SingerTest.java

```
1   package edu.met;
2   import org.springframework.context.*;
3   import org.springframework.context.support.*;
4
5   public class SingerTest {
6
7      private static ApplicationContext ctx;
8      public static void main(String[] args) {
9         // TODO Auto-generated method stub
10
11        ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
12
13        Singer s1=(Singer)ctx.getBean("SingerBean");
14        System.out.println("Id:"+s1.getId());
```

```
15          System.out.println("Name:"+s1.getName());
16      }
17  }
```

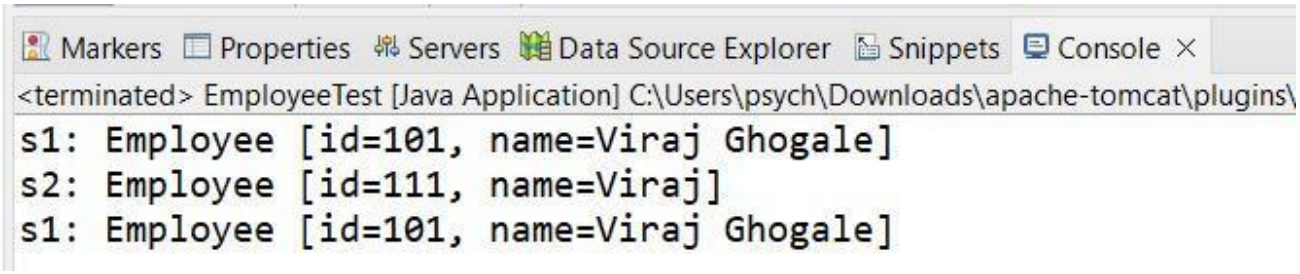
Output:



//2
Source Code:

```
1  package edu.met;
2  import org.springframework.context.*;
3  import org.springframework.context.support.*;
4  public class SingerTest {
5
6     private static ApplicationContext ctx;
7     public static void main(String[] args) {
8        // TODO Auto-generated method stub
9
10       ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
11
12       Singer s1=(Singer)ctx.getBean("SingerBean");
13       System.out.println("s1: "+s1);
14
15       Singer s2=(Singer)ctx.getBean("SingerBean");
16       System.out.println("s2: "+s2);
17    }
18 }
```

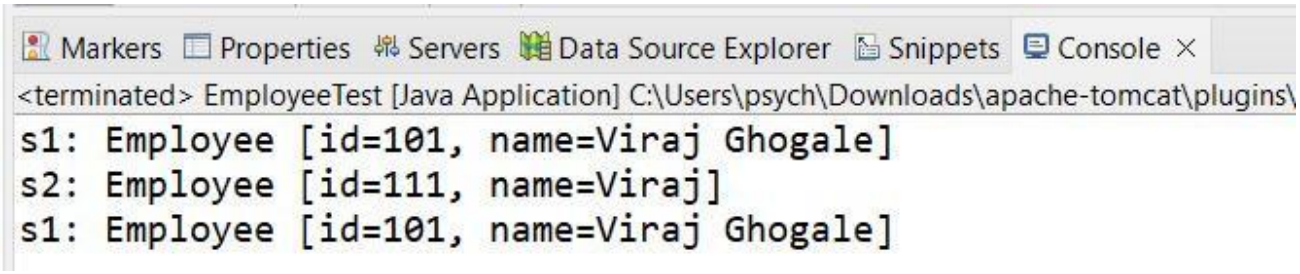
Output:



//3
Source Code:

```
1  package edu.met;
2  import org.springframework.context.*;
3  import org.springframework.context.support.*;
4
5  public class SingerTest {
6
7     private static ApplicationContext ctx;
8     public static void main(String[] args) {
9        // TODO Auto-generated method stub
10
11       ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
12
13       Singer s1=(Singer)ctx.getBean("SingerBean");
14       System.out.println("s1: "+s1);
15
16
17       Singer s2=(Singer)ctx.getBean("SingerBean");
18       s2.setId(102);
19       s2.setName("Ghogale");
20
21       System.out.println("s1: "+s1);
22       System.out.println("s2: "+s2);
23    }
24 }
```

Output:



Program No:	**21**
Roll No :	**1418**
Title of Program :	**Spring**
Objective :	**Employee Application**

Source Code:

Employee.java

```
1  package edu.met;
2  public class Employee {
3  int id;
4  String name;
5  public Employee(int id, String name) {
6  super();
7  this.id = id;
8  this.name = name;
9  }
10 public Employee() {
11 super();
12 // TODO Auto-generated constructor stub
13 }
14 public int getId() {
15 return id;
16 }
17 public void setId(int id) {
18 this.id = id;
19 }
20 public String getName() {
21 return name;
22 }
23 public void setName(String name) {
24 this.name = name;
25 }
26 @Override
27 public String toString() {
28 return "Employee [id=" + id + ", name=" + name + "]";
29 }
30 }
```

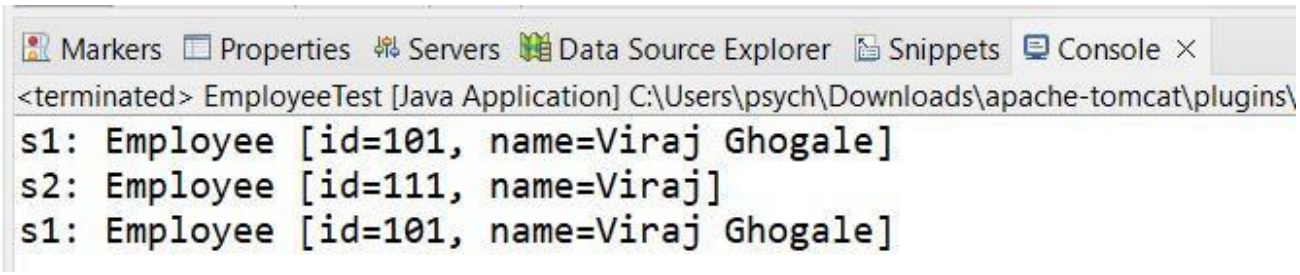
AppCtx.Xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans
3   xmlns="http://www.springframework.org/schema/beans"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xmlns:p="http://www.springframework.org/schema/p"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans
7   http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
8  <bean id="employeeB" class="edu.met.Employee" scope="prototype">
9  <property name = "id" value="101"></property>
10 <property name = "name" value="Viraj "></property>
11 </bean>
12 </beans>
```

EmployeeTest.java

```
1  package edu.met;
2  import org.springframework.context.*;
3  import org.springframework.context.support.*;
4  public class EmployeeTest
5  {
6  private static ApplicationContext ctx;
7  public static void main(String[] args)
8  {
9  ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
10 Employee s1 = (Employee)ctx.getBean("employeeB");
11 System.out.println("s1: " + s1);
12 Employee s2 = (Employee)ctx.getBean("employeeB");
13 s2.setId(111);
14 s2.setName("Viraj");
15 System.out.println("s2: " + s2);
16 System.out.println("s1: " + s1);
17 }
18 }
```

Output:


Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> EmployeeTest [Java Application] C:\Users\psych\Downloads\apache-tomcat\plugins\
s1: Employee [id=101, name=Viraj Ghogale]
s2: Employee [id=111, name=Viraj]
s1: Employee [id=101, name=Viraj Ghogale]

Program No:	**22**
Roll No :	**1418**
Title of Program :	**Spring**
Objective :	**Circular Dependency**

Source Code:

ClassA.java

```
1  package edu.met;
2  public class ClassA {
3   private ClassB classB;
4   public ClassA() {
5   }
6   public ClassB getClassB() {
7  return classB;
8  }
9  public void setClassB(ClassB classB) {
10  this.classB = classB;
11  }
12 }
```

ClassB.java

```
1  package edu.met;
2  public class ClassB {
3   private ClassA classA;
4   public ClassB() {
5   }
6   public ClassA getClassA() {
7  return classA;
8  }
9  public void setClassA(ClassA classA) {
10  this.classA = classA;
11  }
12 }
```

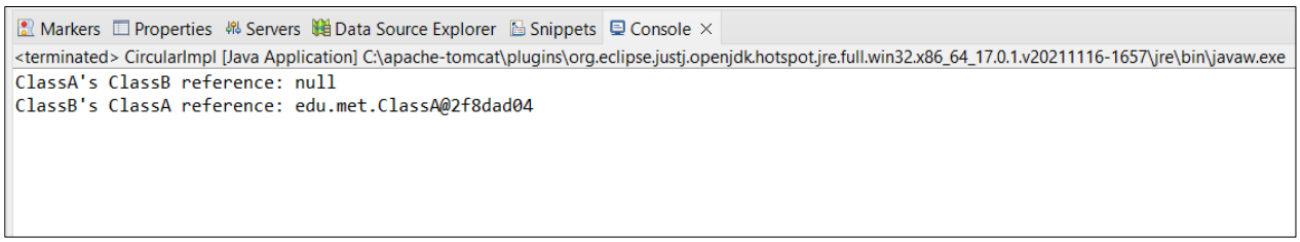
AppCtx.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans
3   xmlns="http://www.springframework.org/schema/beans"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xmlns:p="http://www.springframework.org/schema/p"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans
7   http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
8   <bean id="classA" class="edu.met.ClassA">
9   <!-- Note: No direct reference to classB here -->
10  </bean>
11  <bean id="classB" class="edu.met.ClassB">
12  <property name="classA" ref="classA" />
13  </bean>
14 </beans>
```

CircularImpl.java

```
1  package edu.met;
2  import org.springframework.context.ApplicationContext;
3  import org.springframework.context.support.ClassPathXmlApplicationContext;
4  public class CircularImpl {
5  public static void main(String[] args) {
6  // Load the Spring application context from the configuration file
7   ApplicationContext context = new
8  ClassPathXmlApplicationContext("/resources/AppCtx.xml");
9   // Retrieve instances of ClassA and ClassB from the context
10  ClassA classA = (ClassA) context.getBean("classA");
11  ClassB classB = (ClassB) context.getBean("classB");
12  // Display information to verify the setup
13  System.out.println("ClassA's ClassB reference: " + classA.getClassB());
14  System.out.println("ClassB's ClassA reference: " + classB.getClassA());
15 }
16 }
```

Output:


Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> CircularImpl [Java Application] C:\apache-tomcat\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe
ClassA's ClassB reference: null
ClassB's ClassA reference: edu.met.ClassA@2f8dad04

Program No:	**23**
Roll No :	**1418**
Title of Program :	**Spring**
Objective :	**Spring AOP**

Date:

Source Code:
BankAccount.java

```
1  package edu.met;
2
3  public class BankAccount {
4
5  int accNo;
6  String name;
7  double balance;
8
9  public BankAccount() {
10    super();
11    // TODO Auto-generated constructor stub
12 }
13
14 public BankAccount(int accNo, String name, double balance) {
15    super();
16    this.accNo = accNo;
17    this.name = name;
18    this.balance = balance;
19 }
20 public int getAccNo() {
21    return accNo;
22 }
23 public void setAccNo(int accNo) {
24    this.accNo = accNo;
25 }
26 public String getName() {
27    return name;
28 }
29 public void setName(String name) {
30    this.name = name;
31 }
32 public Double getBalance() {
```

```
33      return balance;
34  }
35  public void setBalance(Double balance) {
36      this.balance = balance;
37  }
38
39  public void credit(double amt)
40  {
41      this.balance+=amt;
42      System.out.println("Credited Succesfully");
43  }
44
45
46  public void debit(double amt)
47  {
48
49      if(balance-amt>0)
50      {
51          this.balance-=amt;
52          System.out.println("Debited Successfully");
53      }
54      else
55      {
56          //System.out.print("Insufficient Funds");
57          throw new RuntimeException();
58      }
59
60
61  }
62
63  @Override
64  public String toString() {
65      return "BankAccount [accNo=" + accNo + ", name=" + name + ", balance="
66  + balance + "]";
67  }
68
    }
```

AppCtx.Xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xmlns:aop="http://www.springframework.org/schema/aop"
5  xsi:schemaLocation="http://www.springframework.org/schema/beans
6  http://www.springframework.org/schema/beans/spring-beans.xsd
7  http://www.springframework.org/schema/aop
8   http://www.springframework.org/schema/aop/spring-aop.xsd">
9  <aop:aspectj-autoproxy />
10
11 <bean id="bankAcc" class="edu.met.BankAccount">
12       <property name="accNo" value="101" />
13       <property name="name" value="sharvari" />
14       <property name="balance" value="2000" />
15
16    </bean>
17
18 </beans>
```

BankAspect.java

```
1  package edu.met;
2  import org.aspectj.lang.annotation.*;
3  @Aspect
4  public class BankAspect {
5
6     //define a joint point
7     @Pointcut("execution(* edu.met.BankAccount.get*(..))")
8     public void getPointCut() {}
9
10    @Before("getPointCut()")
11    public void beforeMethod() {
12       System.out.println("Before Method");
13    }
14
15    @After("execution (* edu.met.BankAccount.set*(..))")
16    public void AfterMethod() {
17       System.out.println("After set Method");
```

```
18      }
19
20      @Pointcut("execution(* edu.met.BankAccount.debit*(..))")
21      public void creditPt() {}
22
23      @AfterThrowing("creditPt()")
24      public void afterThrow()
25      {
26         System.out.println("Insufficient funds");
27      }
28
29      @AfterReturning("creditPt()")
30      public void afterRet()
31      {
32         System.out.println("After Ret Method");
33      }
34
35      @After("creditPt()")
36      public void after()
37      {
38         System.out.println("After Finally");
39      }
40   }
```

BankTest.java

```
1  package edu.met;
2  import org.springframework.context.*;
3  import org.springframework.context.support.*;
4
5  public class BankTest {
6
7     private static ApplicationContext ctx;
8     public static void main(String[] args) {
9        // TODO Auto-generated method stub
10       ctx=new ClassPathXmlApplicationContext("AppCtx.xml");
11       BankAccount b1=(BankAccount) ctx.getBean("bankAcc");
12       System.out.println("Ac No:"+b1.getAccNo());
13       System.out.println("Ac Name:"+b1.getName());
14       System.out.println("Ac Balance:"+b1.getBalance());
15       b1.debit(25000);
16    }
17 }
```

Output:

BankAccount.java | AppCtx.xml | BankAspect.java | BankTest.java

```
@Aspect
public class BankAspect {

    //define a joint point
    @Pointcut("execution(* edu.met.BankAccount.get*(..))")
    public void getPointCut() {}

    @Before("getPointCut()")
    public void beforeMethod() {
        System.out.println("Before Method");

    }

    @After("getPointCut()")
    public void AfterMethod() {
        System.out.println("After Method");

    }

```

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

<terminated> BankTest (6) [AspectJ/Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.

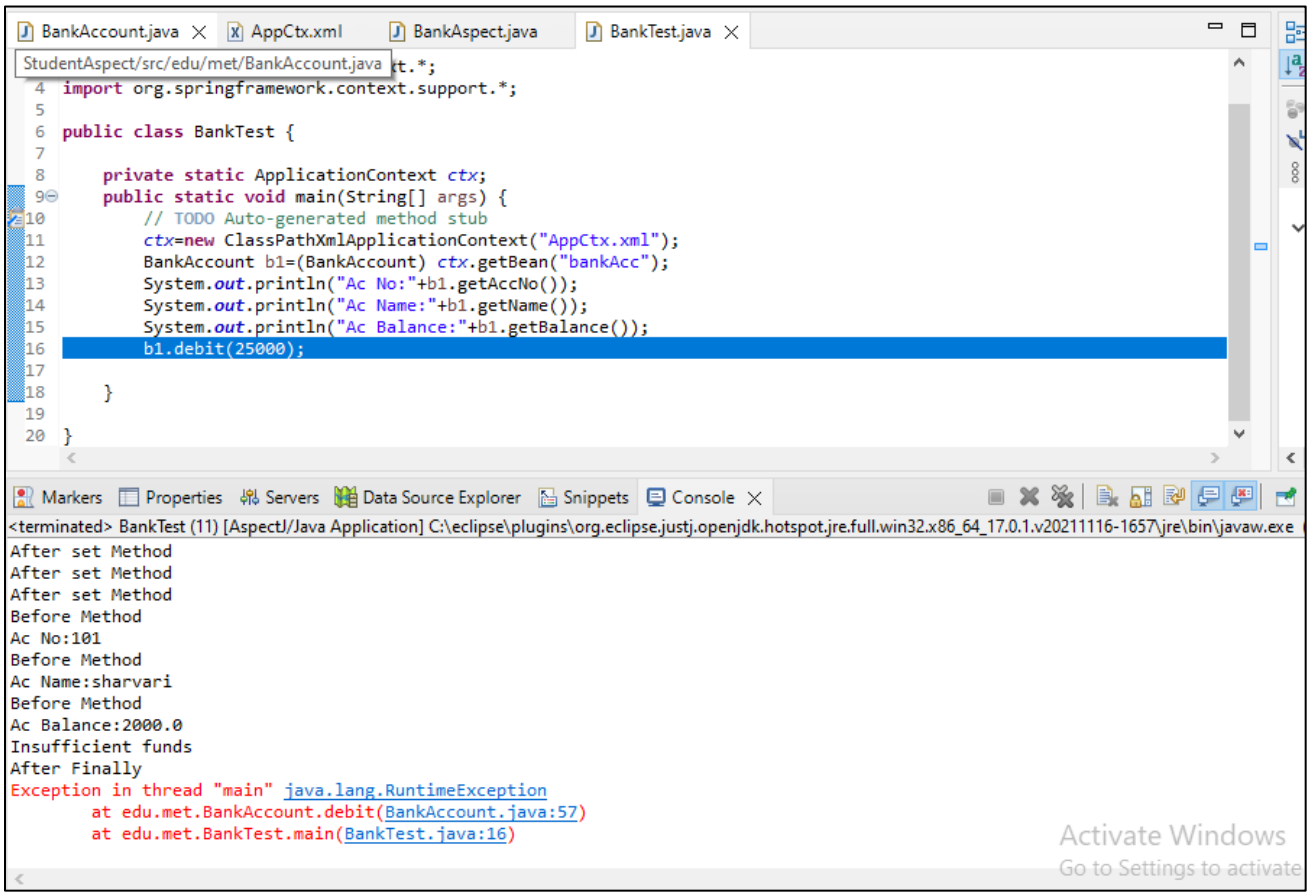
```
Before Method
After Method
Ac No:101
Before Method
After Method
Ac Name:sharvari
Before Method
After Method
Ac Balance:2000.0
```

BankAccount.java | AppCtx.xml | BankAspect.java | BankTest.java

```
1  package edu.met;
2
3  import org.springframework.context.*;
4  import org.springframework.context.support.*;
5
6  public class BankTest {
7
8      private static ApplicationContext ctx;
9      public static void main(String[] args) {
10         // TODO Auto-generated method stub
11         ctx=new ClassPathXmlApplicationContext("AppCtx.xml");
12         BankAccount b1=(BankAccount) ctx.getBean("bankAcc");
13         System.out.println("Ac No:"+b1.getAccNo());
14         System.out.println("Ac Name:"+b1.getName());
15         System.out.println("Ac Balance:"+b1.getBalance());
16         b1.debit(1000);
17
18     }
19
20 }
```

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

```
<terminated> BankTest (10) [AspectJ/Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.e
After set Method
After set Method
After set Method
Before Method
Ac No:101
Before Method
Ac Name:sharvari
Before Method
Ac Balance:2000.0
Debited Successfully
After Ret Method
After Finally
```


BankAccount.java
AppCtx.xml
BankAspect.java
BankTest.java
StudentAspect/src/edu/met/BankAccount.java
import org.springframework.context.support.*;
public class BankTest {
private static ApplicationContext ctx;
public static void main(String[] args) {
// TODO Auto-generated method stub
ctx=new ClassPathXmlApplicationContext("AppCtx.xml");
BankAccount b1=(BankAccount) ctx.getBean("bankAcc");
System.out.println("Ac No:"+b1.getAccNo());
System.out.println("Ac Name:"+b1.getName());
System.out.println("Ac Balance:"+b1.getBalance());
b1.debit(25000);
}
}
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> BankTest (11) [AspectJ/Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe
After set Method
After set Method
After set Method
Before Method
Ac No:101
Before Method
Ac Name:sharvari
Before Method
Ac Balance:2000.0
Insufficient funds
After Finally
Exception in thread "main" java.lang.RuntimeException
at edu.met.BankAccount.debit(BankAccount.java:57)
at edu.met.BankTest.main(BankTest.java:16)
Activate Windows
Go to Settings to activate

Program No:	**24**
Roll No :	**1418**
Title of Program :	**Advanced Spring**
Objective :	**Spring JDBC (Insert Update Delete)**

Source Code:
Movie.java

```
1  package edu.met;
2  public class Movie
3  {
4  int mid;
5  String mname;
6  int duration;
7  public Movie()
8  {
9  super();
10 // TODO Auto-generated constructor stub
11 }
12 public Movie(int mid, String mname, int duration)
13 {
14 super();
15 this.mid = mid;
16 this.mname = mname;
17 this.duration = duration;
18 }
19 public int getMid() {
20 return mid;
21 }
22 public void setMid(int mid) {
23 this.mid = mid;
24 }
25 public String getMname() {
26 return mname;
27 }
28 public void setMname(String mname) {
29 this.mname = mname;
30 }
31 public int getDuration() {
32 return duration;
33 }
34 public void setDuration(int duration) {
```

```
35  this.duration = duration;
36  }
37  }
```

MovieDAO.java

```
1   package edu.met;
2   import java.util.*;
3   import org.springframework.jdbc.core.*;
4   public class MovieDao {
5   JdbcTemplate jdbcT;
6   public MovieDao(JdbcTemplate jdbcT) {
7   super();
8   this.jdbcT = jdbcT;
9   }
10  public int saveMovie(Movie m1) {
11  String insQ = "INSERT INTO movies23
12  VALUES("+m1.getMid()+",'"+m1.getMname()+"',"+m1.getDuration()+")";
13  return jdbcT.update(insQ);
14  }
15  public List<Movie> getAll()
16  {
17  String selQ="SELECT * FROM movies23";
18  return jdbcT.query(selQ, new MovieRowMapper());
19  }
20  }
```

AppCtx.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans
3   xmlns="http://www.springframework.org/schema/beans"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xmlns:p="http://www.springframework.org/schema/p"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans
7   http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
8
9   <bean id="dataSource"
10 class="org.springframework.jdbc.datasource.DriverManagerDataSource">
11  <property name="driverClassName" value="org.postgresql.Driver"/>
12  <property name="url" value="jdbc:postgresql://localhost:5432/postgres"/>
13  <property name="username" value="postgres"/>
14  <property name="password" value="password"/>
15  </bean>
16
17  <bean id="jdbcT" class="org.springframework.jdbc.core.JdbcTemplate">
18  <property name="dataSource" ref="dataSource"/>
19  </bean>
20
21  <bean id="movieD" class="edu.met.MovieDao">
22  <constructor-arg>
23  <ref bean="jdbcT"/>
24  </constructor-arg>
25  </bean>
26
27  <bean id="movie" class="edu.met.Movie">
28  <property name="mid" value="104"/>
29  <property name="mname" value="Pathan"/>
30  <property name="duration" value="156"/>
31  </bean>
32 </beans>
```

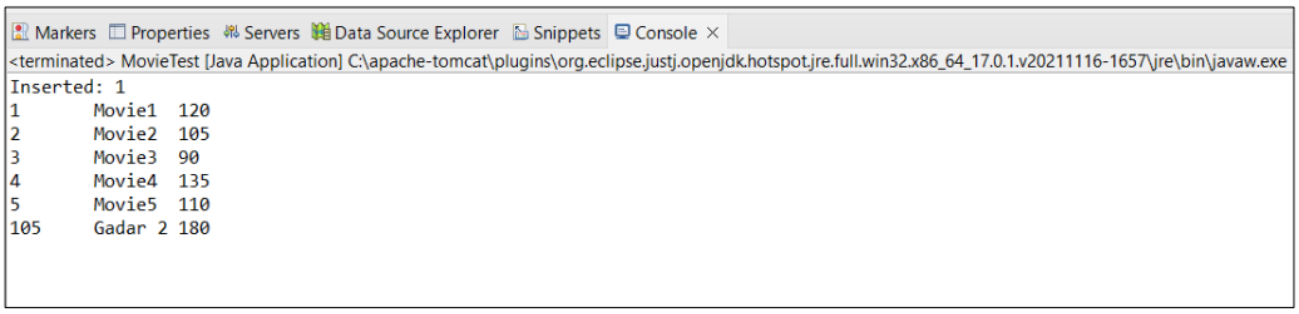
MovieTest.java

```
1  package edu.met;
2  import org.springframework.context.*;
3  import org.springframework.context.support.*;
4  import java.util.*;
5  public class MovieTest {
6  private static ApplicationContext ctx;
7  public static void main(String[] args) {
8  ctx = new ClassPathXmlApplicationContext("AppCtx.xml");
9  Movie m1 = (Movie)ctx.getBean("movie");
10 MovieDao md = (MovieDao)ctx.getBean("movieD");
11 m1.setMid(105);
12 m1.setMname("Gadar 2");
13 m1.setDuration(180);
14 System.out.println("Inserted: " + md.saveMovie(m1));
15 List<Movie> mov = md.getAll();
16 for(Movie m : mov)
17 {
18 System.out.println(m.getMid()+"\t"+m.getMname()+"\t"+m.getDuration());
19 }
20 }
21 }
```

MovieRowMapper.java

```
1  package edu.met;
2  import java.sql.*;
3  import org.springframework.jdbc.core.*;
4  public class MovieRowMapper implements RowMapper<Movie>
5  {
6  @Override
7  public Movie mapRow(ResultSet arg0, int arg1) throws SQLException
8  {
9  Movie m1 = new Movie();
10 m1.setMid(arg0.getInt(1));
11 m1.setMname(arg0.getString(2));
12 m1.setDuration(arg0.getInt(3));
13 return m1;
14 }
15 }
```

Output:


Markers Properties Servers Data Source Explorer Snippets Console
<terminated> MovieTest [Java Application] C:\apache-tomcat\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe
Inserted: 1
1 Movie1 120
2 Movie2 105
3 Movie3 90
4 Movie4 135
5 Movie5 110
105 Gadar 2 180

Program No:	**25**
Roll No :	**1418**
Title of Program :	**Advanced Spring**
Objective :	**Spring Boot Application**

Source Code:

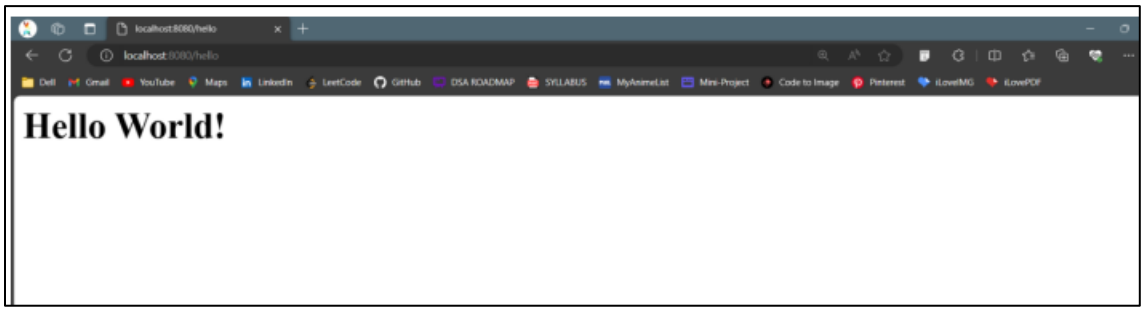
Controller.java

```
1  package com.example.demo;
2  import org.springframework.web.bind.annotation.GetMapping;
3  import org.springframework.web.bind.annotation.RequestMapping;
4  import org.springframework.web.bind.annotation.RestController;
5  @RestController
6  @RequestMapping("/")
7  public class Controller {
8  @GetMapping("/hello")
9  public String sayHello() {
10  return "<h1>Hello World!</h1>";
11 }
12 }
```

SpringBootDemoApplication.java

```
1 package com.example.demo;
2 import org.springframework.boot.SpringApplication;
3 import org.springframework.boot.autoconfigure.SpringBootApplication;
4 @SpringBootApplication
5 public class SpringBootDemoApplication {
6 public static void main(String[] args) {
7 SpringApplication.run(SpringBootDemoApplication.class, args);
8 }
9 }
```

Output:



Program No:	**26**
Roll No :	**1418**
Title of Program :	**Advanced Spring**
Objective :	**Spring Boot Using JDBC**

Source Code:

Pom.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5  https://maven.apache.org/xsd/maven-4.0.0.xsd">
6  <modelVersion>4.0.0</modelVersion>
7  <parent>
8  <groupId>org.springframework.boot</groupId>
9  <artifactId>spring-boot-starter-parent</artifactId>
10 <version>3.2.0</version>
11 <relativePath/> <!-- lookup parent from repository -->
12 </parent>
13 <groupId>com.example</groupId>
14 <artifactId>ProductServiceApp</artifactId>
15 <version>0.0.1-SNAPSHOT</version>
16 <name>ProductServiceApp</name>
17 <description>Demo project for Spring Boot</description>
18 <properties>
19 <java.version>17</java.version>
20 </properties>
21 <dependencies>
22 <dependency>
23 <groupId>org.springframework.boot</groupId>
24 <artifactId>spring-boot-starter-actuator</artifactId>
25 </dependency>
26 <dependency>
27 <groupId>org.springframework.boot</groupId>
28 <artifactId>spring-boot-starter-data-jdbc</artifactId>
29 </dependency>
30 <dependency>
31 <groupId>org.springframework.boot</groupId>
32 <artifactId>spring-boot-starter-data-rest</artifactId>
33 </dependency>
34 <dependency>
```

```
35 <groupId>org.springframework.boot</groupId>
36 <artifactId>spring-boot-starter-web</artifactId>
37 </dependency>
38 <dependency>
39 <groupId>org.springframework.data</groupId>
40 <artifactId>spring-data-rest-hal-explorer</artifactId>
41 </dependency>
42 <dependency>
43 <groupId>org.springframework.boot</groupId>
44 <artifactId>spring-boot-devtools</artifactId>
45 <scope>runtime</scope>
46 <optional>true</optional>
47 </dependency>
48 <dependency>
49 <groupId>com.h2database</groupId>
50 <artifactId>h2</artifactId>
51 <scope>runtime</scope>
52 </dependency>
53 <dependency>
54 <groupId>org.postgresql</groupId>
55 <artifactId>postgresql</artifactId>
56 <scope>runtime</scope>
57 </dependency>
58 <dependency>
59 <groupId>org.springframework.boot</groupId>
60 <artifactId>spring-boot-starter-test</artifactId>
61 <scope>test</scope>
62 </dependency>
63 </dependencies>
64 <build>
65 <plugins>
66 <plugin>
67 <groupId>org.springframework.boot</groupId>
68 <artifactId>spring-boot-maven-plugin</artifactId>
69 </plugin>
70 </plugins>
71 </build>
72 </project>
```

Application.properties

```
spring.datasource.url = jdbc:postgresql://localhost:5432/postgres
spring.datasource.username = postgres
spring.datasource.password = password
```

Product.java

```
package com.example.demo;
public class Product {
int pid;
String pname;
String pdesc;
int pqty;
public Product() {
super();
// TODO Auto-generated constructor stub
}
public Product(int pid, String pname, String pdesc, int pqty) {
super();
this.pid = pid;
this.pname = pname;
this.pdesc = pdesc;
this.pqty = pqty;
}
public int getPid() {
return pid;
}
public void setPid(int pid) {
this.pid = pid;
}
public String getPname() {
return pname;
}
public void setPname(String pname) {
this.pname = pname;
}
public String getPdesc() {
return pdesc;
}
public void setPdesc(String pdesc) {
this.pdesc = pdesc;
```

```
35 }
36 public int getPqty() {
37 return pqty;
38 }
39 public void setPqty(int pqty) {
40 this.pqty = pqty;
41 }
42 }
```

GreetingController.java

```
1  package com.example.demo;
2  import org.springframework.web.bind.annotation.*;
3  @RestController
4  public class GreetingController {
5  @GetMapping("/")
6  public String getWelcome()
7  {
8  return "<HTML><BODY><H1>Welcome to
9  MET</H1></BODY></HTML>";
10 }
11 @GetMapping("/welcome")
12 public String getWelcome2()
13 {
14 return "<HTML><BODY><H1>Welcome to MET!
15 Everyone</H1></BODY></HTML>";
16 }
17 }
```

ProductController.java

```
1  package com.example.demo;
2  import java.util.*;
3  import org.springframework.beans.factory.annotation.Autowired;
4  import org.springframework.web.bind.annotation.GetMapping;
5  import org.springframework.web.bind.annotation.PathVariable;
6  import org.springframework.web.bind.annotation.RequestMapping;
7  import org.springframework.web.bind.annotation.RequestMethod;
8  import org.springframework.web.bind.annotation.RestController;
9  @RestController
10 public class ProductController {
11 @Autowired
12 ProductRepo repo;
13 @GetMapping("/product")
14 public List<Product> getProducts()
15 {
16 return repo.getAll();
17 }
18 @RequestMapping(value="/product/{pid}",method=RequestMethod.DELETE)
19 public String deleteP(@PathVariable("pid") String pid)
20 {
21 repo.deleteProduct(pid);
22 return "<html><body>Object Deleted</body></html>";
23 }
24 }
```

ProductRepo.java

```
1  package com.example.demo;
2  import org.springframework.beans.factory.annotation.Autowired;
3  import org.springframework.jdbc.core.JdbcTemplate;
4  import org.springframework.stereotype.*;
5  import java.util.*;
6  @Repository
7  public class ProductRepo {
8  @Autowired
9  JdbcTemplate jdbcT;
10 public List<Product> getAll()
11 {
12 String selQ = "SELECT * FROM products23";
13 return jdbcT.query(selQ, new ProductRowMapper());
```

```
14 }
15 public int deleteProduct(String pid)
16 {
17 String delQ = "DELETE FROM products23 where pid=?";
18 return jdbcT.update(delQ,Integer.parseInt(pid));
19 }
20 }
```

ProductRowMapper.java

```
1  package com.example.demo;
2  import java.sql.ResultSet;
3  import java.sql.SQLException;
4  import org.springframework.jdbc.core.RowMapper;
5  public class ProductRowMapper implements RowMapper<Product> {
6  @Override
7  public Product mapRow(ResultSet rs, int rowNum) throws SQLException {
8  // TODO Auto-generated method stub
9  Product p1 = new Product();
10 p1.setPid(rs.getInt(1));
11 p1.setPname(rs.getString(2));
12 p1.setPdesc(rs.getString(3));
13 p1.setPqty(rs.getInt(4));
14 return p1;
15 }
16 }
```

ProductServiceAppApplication.java

```
1 package com.example.demo;
2 import org.springframework.boot.SpringApplication;
3 import org.springframework.boot.autoconfigure.SpringBootApplication;
4 @SpringBootApplication
5 public class ProductServiceAppApplication {
6 public static void main(String[] args) {
7 SpringApplication.run(ProductServiceAppApplication.class, args);
8 }
9 }
```

Output:

```
[
    {
        "pid": 1,
        "pname": "ProductA",
        "pdesc": "DescriptionA",
        "pqty": 10
    },
    {
        "pid": 2,
        "pname": "ProductB",
        "pdesc": "DescriptionB",
        "pqty": 20
    },
    {
        "pid": 3,
        "pname": "ProductC",
        "pdesc": "DescriptionC",
        "pqty": 15
    },
    {
        "pid": 4,
        "pname": "ProductD",
        "pdesc": "DescriptionD",
        "pqty": 25
    },
    {
        "pid": 5,
        "pname": "ProductE",
        "pdesc": "DescriptionE",
        "pqty": 30
    }
]
```

Program No:	**27**
Roll No :	**1418**
Title of Program :	**Advanced Spring**
Objective :	**Spring Boot MVC**

Source Code:
User.java

```
1  package com.example.demo;
2  public class User {
3  int id;
4  String name;
5  String email;
6  public User(int id, String name, String email) {
7  super();
8  this.id = id;
9  this.name = name;
10 this.email = email;
11 }
12 public User() {
13 super();
14 // TODO Auto-generated constructor stub
15 }
16 public int getId() {
17 return id;
18 }
19 public void setId(int id) {
20 this.id = id;
21 }
22 public String getName() {
23 return name;
24 }
25 public void setName(String name) {
26 this.name = name;
27 }
28 public String getEmail() {
29 return email;
30 }
31 public void setEmail(String email) {
32 this.email = email;
33 }
```

```
34 }
```

Home.jsp

```
1  <! DOCTYPE html>
2  <%@ page language="java" %>
3  <html>
4  <body>
5  <h1>This is home page</h1>
6  </body>
7  </html>
```

Application.properties

```
1  spring.mvc.view.prefix=/WEB-INF/jsp/
2  spring.mvc.view.suffix=.jsp
```

HomeController.java

```
1  package com.example.demo;
2  import org.springframework.stereotype.Controller;
3  import org.springframework.web.bind.annotation.*;
4  @Controller
5  public class HomeController {
6  @RequestMapping(value="/", method=RequestMethod.GET)
7  public String getRoute()
8  {
9  return "home";
10 }
11 }
```

Pom.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5  https://maven.apache.org/xsd/maven-4.0.0.xsd">
6  <modelVersion>4.0.0</modelVersion>
7  <parent>
8  <groupId>org.springframework.boot</groupId>
9  <artifactId>spring-boot-starter-parent</artifactId>
10 <version>3.2.0</version>
11 <relativePath/> <!-- lookup parent from repository -->
12 </parent>
13 <groupId>com.example</groupId>
14 <artifactId>UserMVCApp</artifactId>
15 <version>0.0.1-SNAPSHOT</version>
16 <name>UserMVCApp</name>
17 <description>Demo project for Spring Boot</description>
18 <properties>
19 <java.version>17</java.version>
20 </properties>
21 <dependencies>
22 <dependency>
23 <groupId>org.springframework.boot</groupId>
24 <artifactId>spring-boot-starter-actuator</artifactId>
25 </dependency>
26 <dependency>
27 <groupId>org.springframework.boot</groupId>
28 <artifactId>spring-boot-starter-web</artifactId>
29 </dependency>
30 <dependency>
31 <groupId>org.springframework.boot</groupId>
32 <artifactId>spring-boot-devtools</artifactId>
33 <scope>runtime</scope>
34 <optional>true</optional>
35 </dependency>
36 MUMBAI EDUCATIONAL TRUST
37 MET Institute of Computer Science
38 95 |
39 <dependency>
40 <groupId>org.springframework.boot</groupId>
41 <artifactId>spring-boot-starter-test</artifactId>
```

```
<scope>test</scope>
</dependency>
<dependency>
 <groupId>org.apache.tomcat.embed</groupId>
 <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

UserMVCApplication.java

```
package com.example.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class UserMvcAppApplication {
public static void main(String[] args) {
SpringApplication.run(UserMvcAppApplication.class, args);
}
}
```

Output:


Dell
Gmail
YouTube
Maps
LinkedIn
LeetCode
GitHub
DSA ROADMAP
SYLLABUS
MyAnimeList
Mini-Project
Code to Image
This is home page