Government of Karnataka
Department of Collegiate Education

Dr. P. DAYANANDA PAI – P. SATHISHA PAI
GOVERNMENT FIRST GRADE COLLEGE CARSTREET MANGALORE

LABORATORY CERTIFICATE

This is to certify that Mr./Ms. _____has satisfactorily completed the practical works in ADVANCED JAVA AND J2EE LAB prescribed by the Mangalore University for SIXTH SEMESTER BCA course, in the laboratory of this college during the year 2024-2025.

_____ _____

Lecturer in charge **Head of the Department**

Name of the Candidate :
Registration Number :
Examination Centre : Dr. PDP-PSP GFGC MANGALORE
Date of Examination : .....

EXAMINERS:

_____ _____

INTERNAL **EXTERNAL**

INDEX

PART A

/* ********** */

PROGRAM: 1

Aim: Write a program to convert numbers into words using Enumerations with constructors, methods and instance variables. (INPUT RANGE-0 TO 99999)
EX: 36 THIRTY SIX

Date: 11/02/2025

/* ********** */

```
package parta1;
import java.util.Scanner;
public class PartA1 {
   enum Units {
      ZERO(""), ONE("ONE"), TWO("TWO"), THREE("THREE"), FOUR("FOUR"),
FIVE("FIVE"), SIX("SIX"), SEVEN("SEVEN"), EIGHT("EIGHT"), NINE("NINE");
      private final String word;
      Units(String word) {
         this.word = word;
      }
      public String getWord() {
         return word;
      }
   }
   enum Tens {
      TEN("TEN"), ELEVEN("ELEVEN"), TWELVE("TWELVE"),
THIRTEEN("THIRTEEN"), FOURTEEN("FOURTEEN"), FIFTEEN("FIFTEEN"),
SIXTEEN("SIXTEEN"), SEVENTEEN("SEVENTEEN"), EIGHTEEN("EIGHTEEN"),
NINETEEN("NINETEEN"), TWENTY("TWENTY"), THIRTY("THIRTY"),
FOURTY("FOURTY"), FIFTY("FIFTY"), SIXTY("SIXTY"), SEVENTY("SEVENTY"),
EIGHTY("EIGHTY"), NINTY("NINTY");
      private final String word;
      Tens(String word) {
         this.word = word;
      }
      public String getWord() {
         return word;
      }
   }
   enum Thousands {
      THOUSAND("THOUSAND");
      private final String word;
      Thousands(String word) {
         this.word = word;
      }
      public String getWord() {
         return word;
      }
   }
   public static String convertToWords(int number) {
      if (number == 0) {
         return "ZERO";
      }
```

```
        String words = " ";
        if (number / 1000 > 0) {
            words += convertThreeDigitsToWords(number / 1000) + " " +
Thousands.THOUSAND.getWord() + " ";
            number %= 1000;
        }
        words += convertThreeDigitsToWords(number);
        return words.trim();
    }
    public static String convertThreeDigitsToWords(int number) {
        String words = " ";
        if (number / 100 > 0) {
            words += Units.values()[number / 100].getWord() + "HUNDRED";
            number %= 100;
        }
        if (number >= 20) {
            words += Tens.values()[(number / 10) + 8].getWord() + " ";
            number %= 10;
        } else if (number >= 10) {
            words += Tens.values()[number % 10].getWord() + " ";
            number = 0;
        }
        if (number > 0) {
            words += Units.values()[number].getWord() + " ";
        }
        return words;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number between 0 and 99999:");
        int number = scanner.nextInt();
        if (number < 0 || number > 99999) {
            System.out.println("Please enter input between 0 and 99999");
        } else {
            String result = convertToWords(number);
            System.out.println(result);
        }
    }
}
```

OUTPUT:

```
Enter a number between 0 and 99999:

20

TWENTY
```

```
Enter a number between 0 and 99999:

212302

Please enter input between 0 and 99999
```

/* ********** */

PROGRAM: 2

Aim: Find the second maximum and second minimum in a set of numbers using auto boxing and unboxing.

Date: 11/02/2025

/* ********** */

```
package parta2;
import java.util.*;
public class PartA2 {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the size of array");
      int n = sc.nextInt();
      List<Integer> arlist = new ArrayList<>();
      int num[] = new int[n];
      System.out.println("Enter a number");
      for (int i = 0; i < n; i++) {
         arlist.add(sc.nextInt());
      }
      Collections.sort(arlist);
      int secondMin = arlist.get(1);
      int secondMax = arlist.get(arlist.size() - 2);
      System.out.println("Second Minimum:" + secondMin);
      System.out.println("Second Maximum:" + secondMax);
   }
}
```

OUTPUT:

```
Enter the size of array
6
Enter a number
30
10
5
42
20
50
Second Minimum:10
Second Maximum:42
```

/* ******** */**

PROGRAM: 3

Aim: Write a menu driven program to create an Arraylist and perform the following operations

i) Adding elements
ii) Sorting elements
iii) Replace an element with another
iv) Removing an element
v) Displaying all the elements
vi) Adding an element between two element

Date: 18/02/2025

/* ******** */**

```
package parta3;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;
public class PartA3 {
   public static void main(String[] args) {
      int choice;
      Scanner in = new Scanner(System.in);
      ArrayList<Integer> alist = new ArrayList<Integer>();
      int val, fval, pos;
      do {
         System.out.println("********MENU********");
         System.out.println("1. Add");
         System.out.println("2. Sort");
         System.out.println("3. Replace");
         System.out.println("4. Remove");
         System.out.println("5. Display");
         System.out.println("6. Add in between");
         System.out.println("7. Exit");
         System.out.println("_____");
         System.out.println("Enter your choice:");
         choice = in.nextInt();
         switch (choice) {
            case 1:
               System.out.println("Enter a number:");
               val = in.nextInt();
               alist.add(val);
               System.out.println("Item added to the list");
               break;
            case 2:
               System.out.println("Sorting");
               Collections.sort(alist);
               System.out.println("Sorting complete");
               break;
            case 3:
               System.out.println("Enter value to find:");
               fval = in.nextInt();
               if (alist.contains(fval)) {
```

```
                System.out.println("Enter the replacement value:");
                val = in.nextInt();
                Collections.replaceAll(alist, fval, val);
                System.out.println("Replacement completed");
             } else {
                System.out.println("Element does not exist");
             }
             break;
          case 4:
             System.out.println("Enter the element to remove:");
             val = in.nextInt();
             if (alist.contains(val)) {
                alist.remove((Integer) val);
                System.out.println(val + " is removed");
             } else {
                System.out.println("Element is not found");
             }
             break;
          case 5:
             System.out.println(alist);
             break;
          case 6:
             System.out.println("Enter the index position:");
             pos = in.nextInt();
             if (pos < alist.size()) {
                System.out.println("Enter the value of new element:");
                val = in.nextInt();
                alist.add(pos, val);
                System.out.println("Element inserted");
             } else {
                System.out.println("Position out of bound");
             }
             break;
          case 7:
             System.out.println("Thank you");
             return;
          default:
             System.out.println("Wrong choice! Try again");
       }
    } while (true);
  }
}
```

OUTPUT:

```
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
5
[]
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
1
Enter a number:
10
Item added to the list
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
1
Enter a number:
40
Item added to the list
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
1
Enter a number:
20
Item added to the list
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
5
[10, 40, 20]
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
2
Sorting
Sorting complete
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
5
[10, 20, 40]
********MENU********
1. Add
2. Sort
3. Replace
```

```
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
3
Enter value to find:
10
Enter the replacement value:
30
Replacement completed
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
5
[30, 20, 40]
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
4
Enter the element to remove:
20
20 is removed
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
```

```
5
[30, 40]
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
6
Enter the index position:
1
Enter the value of new element:
60
Element inserted
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
5
[30, 60, 40]
********MENU********
1. Add
2. Sort
3. Replace
4. Remove
5. Display
6. Add in between
7. Exit
_____
Enter your choice:
8
Wrong choice! Try again
```

/* ********** */

PROGRAM: 4

Aim: Write a java program to find words with even number of characters in a string, then swap the pair of characters in those words and also toggle the characters in a given string.

EX:

Good Morning everyone

Output: oGdo vereoyen

gOOD mORNING EVERYONE

Date: 18/02/2025

/* ********** */

```
package parta4;
import java.util.*;
public class PartA4 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      String str;
      System.out.println("Enter a string:");
      str = scanner.nextLine();
      String punct = " .,!?:;\n\t";
      int start = 0;
      String word = "", revWord = "";
      String togStr = "";
      str = str.trim() + " ";
      for (int i = 0; i < str.length(); i++) {
         if (punct.contains(str.charAt(i) + "")) {
            word = str.substring(start, i);
            start = i + 1;
            StringBuilder sb = new StringBuilder(word.trim());
            char tchar;
            if (sb.length() > 0 && sb.length() % 2 == 0) {
               for (int j = 1; j < sb.length(); j += 2) {
                  tchar = sb.charAt(j);
                  sb.setCharAt(j, sb.charAt(j - 1));
                  sb.setCharAt(j - 1, tchar);
               }
               System.out.println(" " + sb);
            }
            StringBuilder capF = new StringBuilder(word);
            for (int j = 0; j < sb.length(); j++) {
               if (Character.isUpperCase(capF.charAt(j))) {
                  capF.setCharAt(j, Character.toLowerCase(capF.charAt(j)));
               } else if (Character.isLowerCase((capF.charAt(j)))) {
                  capF.setCharAt(j, Character.toUpperCase(capF.charAt(j)));
               }
            }
            togStr += capF;
            togStr += str.charAt(i);
         }
```

```
        }
        System.out.println("\n" + togStr);
    }
}
```

OUTPUT:

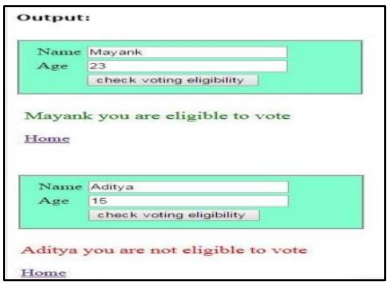
```
Enter a string:
Good Morning everyone
 oGdo
 vereoyen

gOOD mORNING EVERYONE
```

/* ******** */**

PROGRAM: 5

Aim: Write a Servlet program that accepts the age and name and displays if the user is eligible for voting or not



Date: 25/02/2025

/* ******** */**

index.html

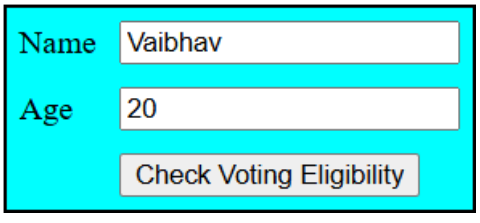
```
<html>
  <head>
    <title>Start Page</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <style>
      table{
        background-color: aqua;
        width: 200px;
        margin-top: 100px;
        margin-left: auto;
        margin-right: auto;
        border: solid 2px;
      }
      td{
        padding: 5px;
      }
    </style>
  </head>
  <body>
    <form method="POST" action="CheckVoter">
      <table>
        <tr>
          <td>Name</td>
          <td><input type="text" name="uname"></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" name="age"></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" name="uname" value="Check Voting
Eligibility"></td>
        </tr>
      </table>
```

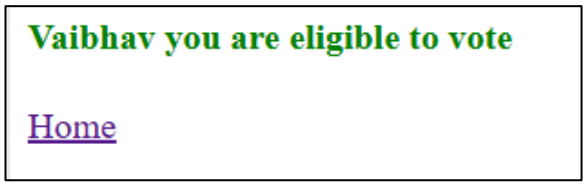
```
        </form>
    </body>
</html>
```

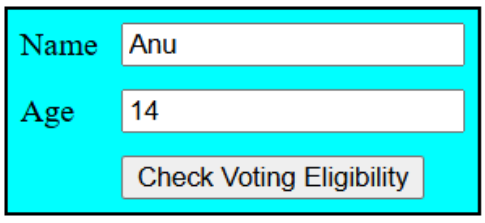
CheckVoter.java

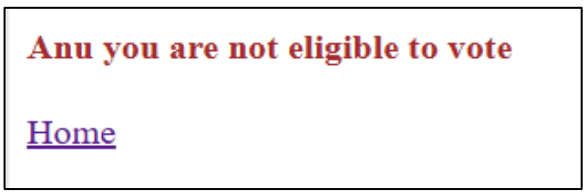
```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class checkVoter extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>checkVoter</title>");
            out.println("</head>");
            out.println("<body>");
            String name = request.getParameter("uname");
            int age = Integer.parseInt(request.getParameter("age"));
            if (age > 18) {
                out.println("<h4 style=\"color:green\">" + name + " you are eligible to
vote</h4>");
            } else {
                out.println("<h4 style=\"color:brown\">" + name + " you are not eligible to
vote</h4>");
            }
            out.println("<a href=\"index.html\">Home</a>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

OUTPUT:









/* ******** */**

PROGRAM: 6

Aim: Write a JSP program to print first 10 Fibonacci and 10 prime numbers.

Date: 25/02/2025

/* ******** */**

FibandPrime.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Fib and Prime</title>
   </head>
   <body>
      <h4>Fibonacci Series</h4>
      <%
         int f1 = 0, f2 = 1, f3, i;
         out.println(f1 + "  " + f2 + "  ");
         for (i = 2; i < 10; i++) {
            f3 = f1 + f2;
            out.println(f3 + "  ");
            f1 = f2;
            f2 = f3;
         }
      %>
      <h4>Prime Numbers</h4>
      <%
         int pn = 2, count = 1;
         boolean isprime;
         while (count <= 10) {
            isprime = true;
            for (i = 2; i <= pn / 2; i++) {
               if (pn % i == 0) {
                  isprime = false;
                  break;
               }
            }
            if (isprime) {
               out.println(pn + "  ");
               count++;
            }
            pn++;
         }
      %>
   </body>
</html>
```

OUTPUT:

```
Fibonacci Series

0 1 1 2 3 5 8 13 21 34

Prime Numbers

2 3 5 7 11 13 17 19 23 29
```

/* ******** */**

PROGRAM: 7

Aim: Write a JSP Program to design a shopping cart to add items, remove item and to display items from the cart using Sessions.

Date: 04/03/2025

/* ******** */**

Item.java

```
package com;
import java.io.Serializable;
public class Item implements Serializable {
   private String name;
   private int qty;
   private double price;
   public Item() {
   }
   public Item(String name, int qty, double price) {
      this.name = name;
      this.qty = qty;
      this.price = price;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public int getQty() {
      return qty;
   }
   public void setQty(int qty) {
      this.qty = qty;
   }
   public double getPrice() {
      return price;
   }
   public void setPrice(double price) {
      this.price = price;
   }
}
```

ShoppingDemo.jsp

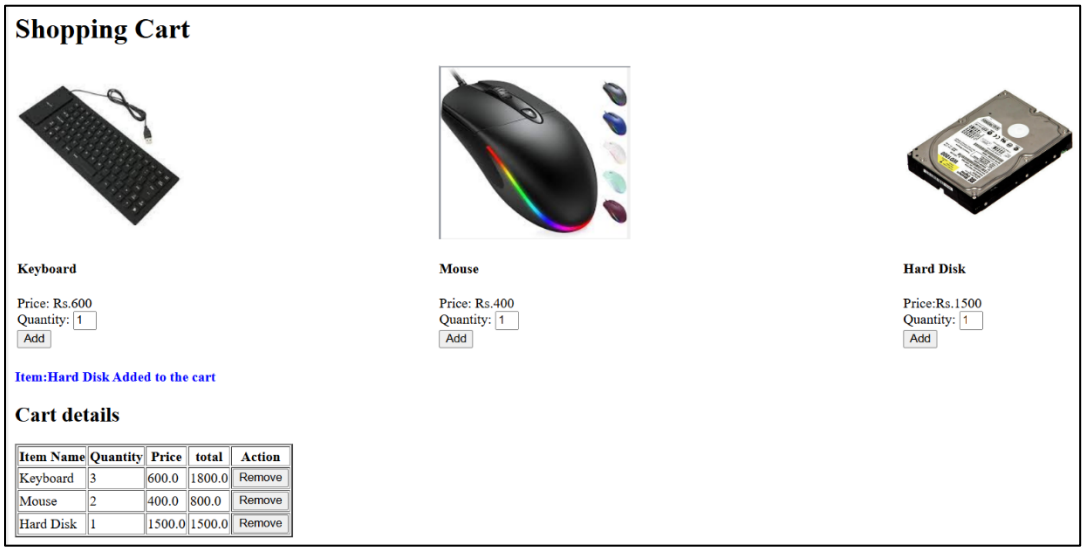
```
<%@page import="com.Item"%>
<%@page import="java.util.ArrayList"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
```

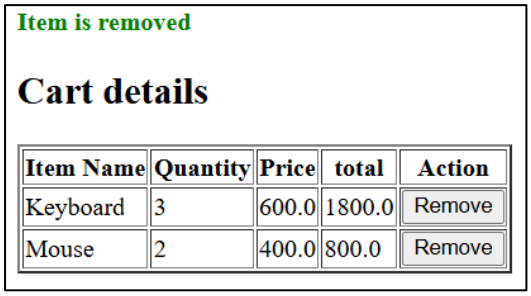
```
<body>
  <h1>Shopping Cart</h1>
  <%
    ArrayList<Item> cart;
    if (request.getSession().getAttribute("cart") == null) {
      cart = new ArrayList<Item>();
      request.getSession().setAttribute("cart", cart);
    } else {
      cart = (ArrayList<Item>) request.getSession().getAttribute("cart");
    }
  %>
  <table width="100%">
    <tr>
      <td>
        <form method="POST">
          <img src="keyboard.jpg" alt="Keyboard" height="200px"/>
          <h4>Keyboard</h4>
          <input type="hidden" value="Keyboard" name="name">
          Price: Rs.600
          <input type="hidden" value="600"name="price">
          <br>
          Quantity:
          <input type="number" name="qty" value="1" style="width:20px">
          <br>
          <input type="submit" name="addBtn" value="Add">
        </form>
      </td>
      <td>
        <form method="POST">
          <img src="mouse.jpg" alt="Mouse" height="200px"/>
          <h4>Mouse</h4>
          <input type="hidden" value="Mouse" name="name">
          Price: Rs.400
          <input type="hidden" value="400" name="price">
          <br>
          Quantity:
          <input type="number" name="qty" value="1" style="width:20px">
          <br>
          <input type="submit" name="addBtn" value="Add">
        </form>
      </td>
      <td>
        <form method="POST">
          <img src="harddisk.jpg" alt="Hard Disk" height="200px"/>
          <h4>Hard Disk</h4>
          <input type="hidden" value="Hard Disk" name="name">
          Price:Rs.1500
          <input type="hidden" value="1500" name="price">
          <br>
          Quantity:
```

```
                <input type="number" name="qty" value="1" style="width:20px">
                <br>
                <input type="submit" name="addBtn" value="Add">
            </form>
        </td>
    </tr>
</table>
<%
    if (request.getParameter("removeBtn") != null) {
        int index = Integer.parseInt(request.getParameter("ino"));
        cart.remove(index);
        out.println("<h4 style=\"color:green\">Item is removed</h4>");
    }
    if (request.getParameter("addBtn") != null) {
        int qty = Integer.parseInt(request.getParameter("qty"));
        if (qty < 0) {
            out.println("<h4 style=\"color:red\">Please enter a positive value for
quantity</h4>");
        } else {
            String name = request.getParameter("name");
            boolean ItemFound = false;
            for (int i = 0; i < cart.size(); i++) {
                Item item1 = cart.get(i);
                if (item1.getName().equals(name)) {
                    item1.setQty(item1.getQty() + qty);
                    out.println("<h4 style=\"color:blue\">Item:" + name + " Added to the
cart</h4>");
                    ItemFound = true;
                    break;
                }
            }
            if (!ItemFound) {
                double price = Double.parseDouble(request.getParameter("price"));
                Item itm = new Item(name, qty, price);
                cart.add(itm);
                out.println("<h4 style=\"color:blue\">Item:" + name + " Added to the
cart</h4>");
            }
        }
    }
    if (cart.size() > 0) {
%>
<h2>Cart details</h2>
<table border="2">
    <tr>
        <th>Item Name</th>
        <th>Quantity</th>
        <th>Price</th>
        <th>total</th>
        <th>Action</th>
```

```
            </tr>
            <%
               for (int i = 0; i < cart.size(); i++) {
                  Item item = cart.get(i);
            %>
            <tr>
               <td><%=item.getName()%></td>
               <td><%=item.getQty()%></td>
               <td><%=item.getPrice()%></td>
               <td><%=item.getQty() * item.getPrice()%></td>
               <td>
                  <form method="POST">
                     <input type="hidden" value="<%=i%>" name="ino">
                     <input type="submit" value="Remove" name="removeBtn">
                  </form>
               </td>
            </tr>
            <%
                  }
               }
            %>
         </table>
         <%
         %>
      </body>
</html>
```

OUTPUT:





/* ******** */**

PROGRAM: 8

Aim: Write a java Servlet program to Download a file and display it on the screen (A link has to be provided in HTML, when the link is clicked corresponding file has to be displayed on screen).

Date: 18/03/2025

/* ******** */**

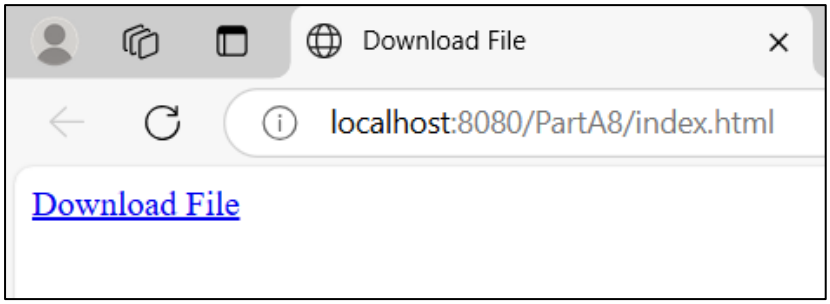
index.html

```
<html>
   <head>
      <title>Download File</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <a href="FileDownloaders?fname=mycv.txt">Download File</a>
   </body>
</html>
```

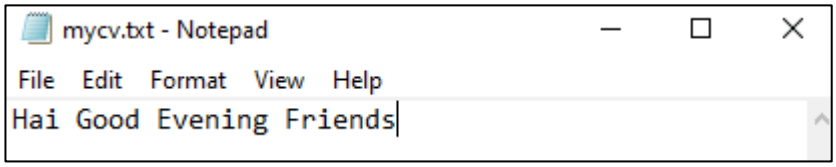
FileDownloaders.java

```
package com;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class FileDownloaders extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/plaintext");
      String fname = request.getParameter("fname");
      response.setContentType("text/plaintext");
      response.setHeader("Content-Disposition", "attachment;filename=\"" + fname +
"\"");
      OutputStream os = response.getOutputStream();
      FileInputStream file = new FileInputStream("C:\\3BCA\\mycv.txt");
      int i = 0;
      while ((i = file.read()) != -1) {
         os.write(i);
      }
      file.close();
      os.close();
   }
}
```

OUTPUT:


Download File
localhost:8080/PartA8/index.html
Download File


Downloads
mycv.txt
Open file


mycv.txt - Notepad
File Edit Format View Help
Hai Good Evening Friends

PART B

/* ******** */**

PROGRAM: 1

Aim: Write a menu driven JDBC program to perform basic operations with Student Table.

MENU

1. **Add new Student**
2. **Delete a specified students Record**
3. **Update Students Address specified students Record**
4. **Search for a particular Student**
5. **Exit**

Student

StRegNo	StName	Stdob	StAddress	StClass	StCourse

Date: 25/03/2025

/* ******** */**

StudentInfoMgt.java

```
package studentinfomgt;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Date;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
public class StudentInfoMgt {
   public static void main(String[] args) {
      try {
         Scanner in = new Scanner(System.in);
         int choice;
         int regno;
         String sname;
         String sadd, sclass, scourse, sql;
         Date dob = null;
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Student", "bcab", "bcab");
         Statement stmt = con.createStatement();
         ResultSet rs;
         do {
            System.out.println("Menu");
            System.out.println("--------------------");
            System.out.println("1. Add Student");
            System.out.println("2. Delete Student");
            System.out.println("3. Update Student");
            System.out.println("4. Search Student");
            System.out.println("5. Exit");
```

```
            System.out.println("--------------------");
            System.out.println("Enter your choice:");
            choice = in.nextInt();
            switch (choice) {
               case 1:
                  System.out.println("----Enter student details----");
                  System.out.println("Reg no:");
                  regno = in.nextInt();
                  System.out.println("Name:");
                  sname = in.next();
                  System.out.println("DOB[yyyy-mm-dd]:");
                  dob = Date.valueOf(in.next());
                  System.out.println("Address:");
                  sadd = in.next();
                  System.out.println("Class:");
                  sclass = in.next();
                  System.out.println("Course:");
                  scourse = in.next();
                  sql = "INSERT INTO BCAB.STDTABLE(STREGNO, STNAME, STDOB,
STADDRESS,STCLASS,STCOURSE ) VALUES(" + regno + ",'" + sname + "','" +
dob.toString() + "','" + sadd + "','" + sclass + "','" + scourse + "')";
                  int result = stmt.executeUpdate(sql);
                  if (result == 1) {
                     System.out.println("Student details are saved");
                  } else {
                     System.out.println("Error while saving student details");
                  }
                  break;
               case 2:
                  System.out.println("Enter Student Regno:");
                  regno = in.nextInt();
                  sql = "SELECT COUNT(*)FROM BCAB.STDTABLE WHERE
STREGNO=" + regno;
                  rs = stmt.executeQuery(sql);
                  rs.next();
                  if (rs.getInt(1) == 1) {
                     sql = "DELETE FROM BCAB.STDTABLE WHERE STREGNO=" +
regno;
                     int res = stmt.executeUpdate(sql);
                     if (res == 1) {
                        System.out.println("Student record is deleted");
                     } else {
                        System.out.println("Record not deleted");
                     }
                  } else {
                     System.out.println("Student record not found");
                  }
                  break;
               case 3:
                  System.out.println("Enter Student Regno:");
```

```
                regno = in.nextInt();
                sql = "SELECT COUNT(*)FROM BCAB.STDTABLE WHERE
STREGNO=" + regno;
                rs = stmt.executeQuery(sql);
                rs.next();
                if (rs.getInt(1) == 1) {
                   sql = "SELECT STADDRESS FROM BCAB.STDTABLE WHERE
STREGNO=" + regno;
                   rs = stmt.executeQuery(sql);
                   rs.next();
                   System.out.println("Old Address:" + rs.getString(1));
                   System.out.println("Enter new address:");
                   String add = in.next();
                   sql = "UPDATE BCAB.STDTABLE SET STADDRESS='" + add + "'
WHERE STREGNO=" + regno;
                   if (stmt.executeUpdate(sql) == 1) {
                      System.out.println("Address updated");
                   } else {
                      System.out.println("Error while updating address");
                   }
                } else {
                   System.out.println("Student record not found");
                }
                break;
             case 4:
                System.out.println("Enter Student Regno:");
                regno = in.nextInt();
                sql = "SELECT * FROM BCAB.STDTABLE WHERE STREGNO=" +
regno;
                rs = stmt.executeQuery(sql);
                if (rs.next()) {
                   System.out.println("--------------------");
                   System.out.println("Student details are");
                   System.out.println("Reg no:" + rs.getInt(1));
                   System.out.println("Name:" + rs.getString(2));
                   System.out.println("DOB:" + rs.getString(3));
                   System.out.println("Address:" + rs.getString(4));
                   System.out.println("Class:" + rs.getString(5));
                   System.out.println("Course:" + rs.getString(6));
                   System.out.println("--------------------");
                }
                break;
             case 5:
                stmt.close();
                con.close();
                System.out.println("Thank you");
                return;
             default:
                System.out.println("Wrong choice \n Try Again");
          }
```

```
            } while (true);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(StudentInfoMgt.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(StudentInfoMgt.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

OUTPUT:

```
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
1
----Enter student details----
Reg no:
101
Name:
Rakesh
DOB[yyyy-mm-dd]:
2004-05-25
Address:
Mangalore
Class:
III
Course:
BCA
Student details are saved
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
1
----Enter student details----
Reg no:
102
Name:
Anu
DOB[yyyy-mm-dd]:
2003-09-21
Address:
Kasaragod
Class:
III
Course:
BSC
Student details are saved
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
1
----Enter student details----
Reg no:
103
Name:
Jithesh
DOB[yyyy-mm-dd]:
2006-02-28
Address:
Bantwal
Class:
I
Course:
BCOM
Student details are saved
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
```

```
Enter your choice:
2
Enter Student Regno:
101
Student record is deleted
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
3
Enter Student Regno:
103
Old Address:Bantwal
Enter new address:
Surathkal
Address updated
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
4
Enter Student Regno:
103
------------------
```

```
Student details are
Reg no:103
Name:Jithesh
DOB:2006-02-28
Address:Surathkal
Class:I
Course:BCOM
------------------
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
6
Wrong choice
 Try Again
Menu
------------------
1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. Exit
------------------
Enter your choice:
5
Thank you
```

Data in Database

SELECT * FROM BCAB.STDTAB... X

Max. rows: 100 Fetched Rows: 2

#	STREGNO	STNAME	STDOB	STADDRESS	STCLASS	STCOURSE
1	101	Rakesh	2004-05-25	Mangalore	III	BCA
2	102	Anu	2003-09-21	Kasaragod	III	BSC

/* ******** */**

PROGRAM: 2

Aim: Write a menu driven JDBC program to perform basic operations with Bank Table.

MENU

1. **Add new Account Holder information.**
2. **Amount Deposit**
3. **Amount Withdrawal (Maintain minimum balance 500 Rs)**
4. **Display all information**
5. **Exit**

Bank

ACC_NO	ACC_NAME	ACC_ADDRESS	BALANCE

Date: 01/04/2025

/* ******** */**

BankAccount.java

```
package bankaccount;
import java.sql.*;
import java.util.*;
public class BankAccount {
   public static void main(String[] args) {
      Scanner in = new Scanner(System.in);
      Connection con;
      int acc_no;
      String name;
      Statement stmt;
      ResultSet result;
      String sql = "";
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         con = DriverManager.getConnection("jdbc:derby://localhost:1527/Student", "bcab",
"bcab");
         stmt = con.createStatement();
         while (true) {
            System.out.print("\n*******Transaction Menu******");
            System.out.print("\n1.Add Account");
            System.out.print("\n2.Deposit");
            System.out.print("\n3.Withdraw");
            System.out.print("\n4.Display");
            System.out.print("\n5.Exit");
            System.out.print("\nEnter the choice:");
            int ch = in.nextInt();
            switch (ch) {
               case 1:
                  System.out.println("Enter the Account holders Name:");
                  name = in.next();
                  System.out.println("Enter the Account number:");
```

```
                acc_no = in.nextInt();
                System.out.println("Enter the address of the account holder:");
                String address = in.next();
                System.out.println("Enter the balance amount");
                float bal = in.nextFloat();
                sql = "INSERT INTO
BANK(ACC_NO,ACC_NAME,ACC_ADDRESS,BALANCE)values(" + acc_no + ",'" +
name + "','" + address + "'," + bal + ")";
                stmt.executeUpdate(sql);
                break;
             case 2:
                System.out.println("Enter the account number:");
                acc_no = in.nextInt();
                System.out.print("\nEnter the amount to be deposited:");
                float d = in.nextFloat();
                if (d <= 0) {
                   System.out.print("\nEnter proper amount.");
                } else {
                   sql = "UPDATE BANK SET BALANCE=BALANCE + " + d + " WHERE
ACC_NO=" + acc_no;
                }
                stmt.executeUpdate(sql);
                break;
             case 3:
                System.out.println("Enter the account number:");
                acc_no = in.nextInt();
                System.out.print("\nEnter the amount to be withdrawn:");
                float w = in.nextFloat();
                if (w <= 0) {
                   System.out.print("\nEnter proper amount.");
                } else {
                   sql = "UPDATE BANK SET BALANCE=BALANCE - " + w + " WHERE
ACC_NO=" + acc_no + " and BALANCE-" + w + ">500";
                   int r = stmt.executeUpdate(sql);
                   if (r == 1) {
                      System.out.println("Updated successfully!!.");
                   } else {
                      System.out.println("Cannot withdraw the amount-LOW BALANCE!!");
                   }
                }
                break;
             case 4:
                result = stmt.executeQuery("SELECT * FROM BANK");
                System.out.println("\n\nACC_NO\t\tName\t\tAddress\t\tBalance");
                System.out.println("-----------------------------------------------------");
                while (result.next()) {
                   System.out.print((int) result.getInt(1) + "\t\t");
                   System.out.print(result.getString(2) + "\t\t");
                   System.out.print(result.getString(3) + "\t\t");
                   System.out.print((int) result.getInt(4) + "\t\t");
```

```
                    System.out.print("\n");
                }
                break;
            case 5:
                System.out.print("\nQuiting.....Thank you");
                System.exit(0);
                break;
            default:
                System.out.print("\nEnter proper choice..");
            }
            System.out.print("\n-------------------------------------------------");
        }
    } catch (Exception se) {
        System.out.println("Exception occurred while getting connection!!!!");
        se.printStackTrace();
    }
  }
}
```

OUTPUT:

```
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:1
Enter the Account holders Name:
Diya
Enter the Account number:
101
Enter the address of the account holder:
Bantwal
Enter the balance amount
1000
-----------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:4
ACC_NO   Name  Address    Balance
-----------------------------------------------
101          Diya    Bantwal    1000
-----------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
```

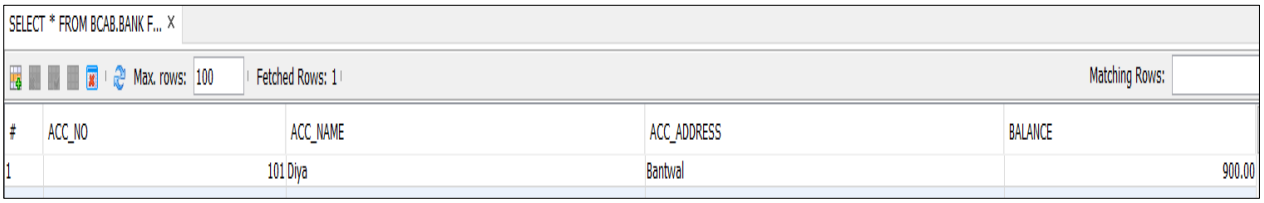
```
4.Display
5.Exit
Enter the choice:2
Enter the account number:
101
Enter the amount to be deposited:500
-----------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:4
ACC_NO   Name  Address    Balance
-----------------------------------------------
101          Diya    Bantwal    1500
-----------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:3
Enter the account number:
101
Enter the amount to be withdrawn:2000
Cannot withdraw the amount-LOW
BALANCE!!
```

```
-------------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:3
Enter the account number:
101
Enter the amount to be withdrawn:600
Updated successfully!!.

-------------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:4
```

```
ACC_NO   Name  Address    Balance
-------------------------------------------------
101          Diya    Bantwal    900
-------------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:8
Enter proper choice..
-------------------------------------------------
*******Transaction Menu******
1.Add Account
2.Deposit
3.Withdraw
4.Display
5.Exit
Enter the choice:5
Quiting.....Thank yo
```

Data in Database



/* ******** */**

PROGRAM: 3

Aim: Write a Java class called Tax with methods for calculating Income Tax. Have this class as a servant and create a server program and register in the rmiregistry. Write a client program to invoke these remote methods of the servant and do the calculations. Accept inputs interactively.

< ₹ 3,00,000	**No Tax**
₹ 3,00,001 to ₹ 6,00,000	**5%**
₹ 6,00,001 to ₹ 9,00,000	**10%**
₹ 9,00,001 to ₹ 12,00,000	**15%**
₹ 12,00,001 to ₹ 15,00,000	**20%**
>₹ 15,00,000	**30%**

Date: 08/04/2025

/* ******** */**

Tax.java

```
package incometax;
import java.rmi.*;
public interface Tax extends Remote
{
   double calTax(double a) throws RemoteException;
}
```

TaxImpl.java

```
package incometax;
import java.rmi.server.*;
import java.rmi.*;
public class TaxImpl extends UnicastRemoteObject implements Tax {
   public TaxImpl() throws RemoteException
   {
   }
   public double calTax(double income) throws RemoteException
   {
      double t;
      if (income <= 300000) {
         t = 0.0;
      } else if (income > 300001 && income <= 600000) {
         t = 0.05f * (income - 300000);
      } else if (income > 600001 && income <= 900000) {
         t = 0.10f * (income - 600000);
      } else if (income > 900001 && income <= 1200000) {
         t = 0.15f * (income - 900000);
      } else if (income > 1200001 && income <= 1500000) {
         t = 0.20f * (income - 1200000);
```

```
        } else {
            t = 0.30f * (income - 1500000);
        }
        return t;
    }
}
```

TaxServer.java

```
package incometax;
import java.rmi.*;
import java.rmi.registry.*;
public class TaxServer {
    public static void main(String args[]) throws RemoteException{
        try{
            TaxImpl timpl = new TaxImpl();
            Registry reg = LocateRegistry.createRegistry(18888);
            reg.rebind("TaxServer", timpl);
            System.out.println("Server is running.....");
        }
        catch(RemoteException e){
            System.out.println("Exception in Server!!....");
        }
    }
}
```

TaxClient.java

```
package incometax;
import java.rmi.registry.*;
import java.util.*;
public class TaxClient {
    public static void main(String args[]){
        try{
            double d;
            Scanner sc = new Scanner(System.in);
            Registry reg = LocateRegistry.getRegistry(18888);
            Tax t = (Tax)reg.lookup("TaxServer");
            System.out.println("Enter the income:");
            d = sc.nextDouble();
            System.out.println("The calculated tax amount is="+t.calTax(d));
        }
        catch(Exception e){
            System.out.println("Exception in Client!!....");
        }
    }
}
```

OUTPUT:

```
Enter the income:
500000
The calculated tax amount is=10000.000149011612
```

/* ******** */**

PROGRAM: 4

Aim: Write a Java class called SimpleInterest with methods for calculating simple interest. Have this class as a servant and create a server program and register in the rmiregistry. Write a client program to invoke these remote methods of the servant and do the calculations. Accept inputs at command prompt.

Date: 15/04/2025

/* ******** */**

SimpleInterest.java

```
package simpleintrest;
import java.rmi.*;
public interface SimpleInterest extends Remote {
   double computeSI(double p, double t, double r) throws RemoteException;
}
```

SimpleInterestImpl.java

```
package simpleintrest;
import java.rmi.*;
import java.rmi.server.*;
public class SimpleInterestImpl extends UnicastRemoteObject implements SimpleInterest {
   public SimpleInterestImpl() throws RemoteException {
      super();
   }
   @Override
   public double computeSI(double p, double t, double r) throws RemoteException {
      return (p * t * r / 100); //To change body of generated methods, choose Tools |
Templates.
   }
}
```

SimpleInterestServer.java

```
package simpleintrest;
import java.rmi.*;
import java.rmi.registry.*;
public class SimpleInterestServer
{
   public static void main(String[] args) throws RemoteException, AlreadyBoundException
  {
      SimpleInterestImpl si = new SimpleInterestImpl();
      Registry reg = LocateRegistry.createRegistry(18888);
      reg.bind("SI", si);
      System.out.println("Server is started.....");
   }
}
```

SimpleInterestClient.java

```
package simpleintrest;
import java.rmi.*;
import java.rmi.registry.*;
import java.util.Scanner;
public class SimpleInterestClient {
   public static void main(String[] args) throws RemoteException, NotBoundException {
      Registry reg = LocateRegistry.getRegistry(18888);
      SimpleInterest si = (SimpleInterest) reg.lookup("SI");
      Scanner sc = new Scanner(System.in);
      double p, t, r;
      String ans = "n";
      do {
         System.out.println("Simple Interest Calculation");
         System.out.println("Principal:");
         p = sc.nextDouble();
         System.out.println("Time:");
         t = sc.nextDouble();
         System.out.println("Rate:");
         r = sc.nextDouble();
         System.out.println("Simple Interest is " + si.computeSI(p, t, r));
         System.out.println("Do you want to continue[Y/N]?");
         sc.nextLine();
         ans = sc.nextLine();
      } while (ans.toLowerCase().charAt(0) == 'y');
   }
}
```

OUTPUT:

```
Simple Interest Calculation
Principal:
1000
Time:
5
Rate:
4
Simple Interest is 200.0
Do you want to continue[Y/N]?
N
```

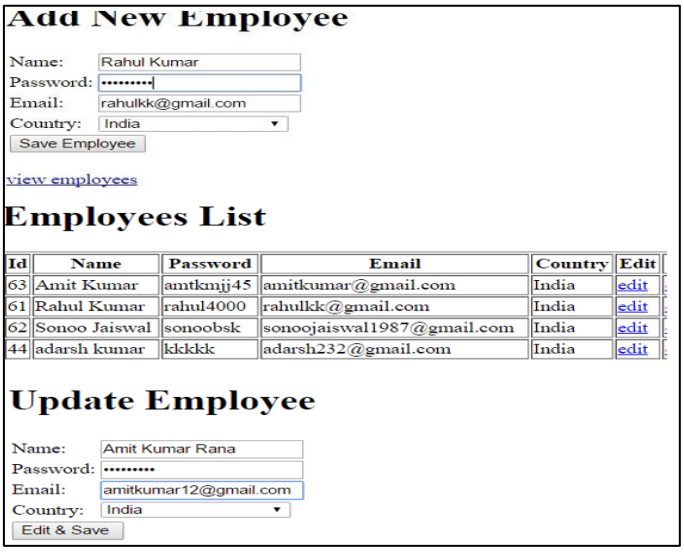
/* ******** */**

PROGRAM: 5

Aim: Write a Servlet Program to perform Insert, update and View operations Employee Table

Employee

Name	Password	Email	Country



Date: 15/04/2025

/* ******** */**

```
CREATE TABLE EMPLOYEE(
id INTEGER PRIMARY KEY GENERATED ALWAYS AS
IDENTITY(START WITH 1, INCREMENT BY 1),
Ename VARCHAR(25),
Password VARCHAR(25),
Email VARCHAR(25),
Country VARCHAR(25)
);
```

index.html

```
<html>
   <head>
      <title>Employee List</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <h1>Add Employee</h1>
      <form method="POST" action="AddEmployee">
         <table>
            <tr>
               <td>Name:</td>
               <td><input type="text" name="ename"></td>
            </tr>
            <tr>
               <td>Password:</td>
               <td><input type="password" name="password"></td>
```

```
            </tr>
            <tr>
               <td>Email:</td>
               <td><input type="email" name="email"></td>
            </tr>
            <tr>
               <td>Country:</td>
               <td>
                  <select name="country">
                     <option>India</option>
                     <option>Nepal</option>
                     <option>China</option>
                     <option>Sri Lanka</option>
                  </select>
               </td>
            </tr>
            <tr>
               <td></td>
               <td colspan="2"><input type="submit" name="submit" value="Save
Employee"></td>
            </tr>
         </table>
      </form>
      <a href="ViewEmployee">View Employee</a>
   </body>
</html>
```

AddEmployee.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class AddEmployee extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try (PrintWriter out = response.getWriter()) {
         String ename = request.getParameter("ename");
         String password = request.getParameter("password");
         String email = request.getParameter("email");
         String country = request.getParameter("country");
         Class.forName("org.apache.derby.jdbc.ClientDriver");
```

```
        Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Student", "bcab",
"bcab");
        Statement stmt = con.createStatement();
        String sql = "INSERT INTO BCAB.EMPLOYEE(ENAME, PASSWORD,
EMAIL,COUNTRY)VALUES('" + ename + "','" + password + "','" + email + "','" +
country + "')";
        int rcount = stmt.executeUpdate(sql);
        stmt.close();
        con.close();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>AddEmployee</title>");
        out.println("</head>");
        out.println("<body>");
        if (rcount == 1) {
           out.println("<h1>Record Saved</h1>");
        } else {
           out.println("<h1>There was an error</h1>");
        }
        out.println("<a href=\"index.html\">Home</a>");
        out.println("</body>");
        out.println("</html>");
      } catch (ClassNotFoundException ex) {
        Logger.getLogger(AddEmployee.class.getName()).log(Level.SEVERE, null, ex);
      } catch (SQLException ex) {
        Logger.getLogger(AddEmployee.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
}
```

ViewEmployee.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class ViewEmployee extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
```

```
        try (PrintWriter out = response.getWriter()) {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Student", "bcab",
"bcab");
            Statement stmt = con.createStatement();
            String sql = "SELECT *FROM BCAB.EMPLOYEE";
            ResultSet rs = stmt.executeQuery(sql);
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ViewEmployee</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Employee List</h1>");
            out.println("<table border=\"2\">");
            out.println("<tr>");
            out.println("<th>id</th>");
            out.println("<th>Name</th>");
            out.println("<th>Password</th>");
            out.println("<th>Email</th>");
            out.println("<th>Country</th>");
            out.println("</tr>");
            while (rs.next()) {
                out.println("<tr>");
                out.println("<td>" + rs.getString("ID") + "</td>");
                out.println("<td>" + rs.getString("ENAME") + "</td>");
                out.println("<td>" + rs.getString("PASSWORD") + "</td>");
                out.println("<td>" + rs.getString("EMAIL") + "</td>");
                out.println("<td>" + rs.getString("COUNTRY") + "</td>");
                out.println("<td> <a href=\"UpdateEmployee?id=" + rs.getString("ID") +
"\">Edit</a></td>");
                out.println("</tr>");
            }
            out.println("</table>");
            out.println("<a href=\"index.html\">Add new record</a>\n" + "");
            out.println("</body>");
            out.println("</html>");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(ViewEmployee.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(ViewEmployee.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

UpdateEmployee.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
```

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class UpdateEmployee extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try (PrintWriter out = response.getWriter()) {
         int id = Integer.parseInt(request.getParameter("id"));
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Student", "bcab",
"bcab");
         Statement stmt = con.createStatement();
         String sql = "SELECT *FROM BCAB.EMPLOYEE WHERE ID=" + id;
         ResultSet rs = stmt.executeQuery(sql);
         rs.next();
         out.println("<!DOCTYPE html>");
         out.println("<html>");
         out.println("<head>");
         out.println("<title>Servlet UpdateEmployee</title>");
         out.println("</head>");
         out.println("<body>");
         out.println("<h1>UpdateEmployee</h1>");
         out.println("<form  method=\"POST\" action=\"SaveEmployee\">");
         out.println("<input type=\"hidden\" name=\"id\" value=\"" + id + "\">");
         out.println("<table>");
         out.println("<tr>");
         out.println("<td>Name:</td>");
         out.println("<td><input type=\"text\" name=\"ename\" value=" +
rs.getString("ENAME") + "></td>");
         out.println("</tr>");
         out.println("<tr>");
         out.println("<td>Password:</td>");
         out.println("<td><input type=\"password\" name=\"password\" value=" +
rs.getString("PASSWORD") + "></td>");
         out.println("</tr>");
         out.println("<tr>");
         out.println("<td>Email:</td>");
         out.println("<td><input type=\"email\" name=\"email\" value=" +
rs.getString("EMAIL") + "></td>");
         out.println("</tr>");
         out.println("<tr>");
```

```
            out.println("<td>Country:</td>");
            out.println("<td><select name=\"country\">");
            String sel = rs.getString("COUNTRY").equals("India") ? "Selected" : "";
            out.println("<option " + sel + ">India</option>");
            sel = rs.getString("COUNTRY").equals("Nepal") ? "Selected" : "";
            out.println("<option " + sel + ">Nepal</option>");
            sel = rs.getString("COUNTRY").equals("China") ? "Selected" : "";
            out.println("<option " + sel + ">China</option>");
            sel = rs.getString("COUNTRY").equals("Sri Lanka") ? "Selected" : "";
            out.println("<option " + sel + ">Sri Lanka</option>");
            out.println("</select></td>");
            out.println("</tr>");
            out.println("<tr>");
            out.println("<td></td>");
            out.println("<td colspan=\"2\"><input type=\"submit\" name=\"submit\"
value=\"Edit Employee\"></td>");
            out.println("</tr>");
            out.println("</table>");
            out.println("");
            out.println("</form>");
            out.println("<a href=\"ViewEmployee\">View Employees</a>");
            out.println("</body>");
            out.println("</html>");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(UpdateEmployee.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(UpdateEmployee.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

SaveEmployee.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class SaveEmployee extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
```

```
            String id = request.getParameter("id");
            String ename = request.getParameter("ename");
            String password = request.getParameter("password");
            String email = request.getParameter("email");
            String country = request.getParameter("country");
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/Student", "bcab",
"bcab");
            Statement stmt = con.createStatement();
            String sql = "UPDATE BCAB.EMPLOYEE SET ENAME='" + ename +
"',PASSWORD='" + password + "',EMAIL='" + email + "',COUNTRY='" + country +
"'WHERE ID=" + id;
            int rcount = stmt.executeUpdate(sql);
            stmt.close();
            con.close();
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet SaveEmployee</title>");
            out.println("</head>");
            out.println("<body>");
            if (rcount == 1) {
               out.println("<h1>Record Saved</h1>");
            } else {
               out.println("<h1>There was an error</h1>");
            }
            out.println("<a href=\"ViewEmployee\">View Employees</a>");
            out.println("</body>");
            out.println("</html>");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(SaveEmployee.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(SaveEmployee.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

OUTPUT:

Employee List

id	Name	Password	Email	Country	
1	Dhanush	dhanu@123	dhanush@gmail.com	India	Edit

Add new record

UpdateEmployee

Name: Dhanush A
Password: ••••••••••
Email: dhanush123@gmail.com
Country: India
Edit Employee

View Employees

Record Saved

View Employees

Employee List

id	Name	Password	Email	Country	
1	Dhanush A	dhanu@123	dhanush123@gmail.com	India	Edit

Add new record

/* ******** */**

PROGRAM: 6

Aim: Write a java JSP program to get student information through a HTML and create a JAVA Bean Class, populate Bean and Display the same information through another JSP

Date: 22/04/2025

/* ******** */**

Student1.java

```
package com;
import java.io.Serializable;
public class Student1 implements Serializable {
   private String regNo;
   private String name;
   private String course;
   private String sem;
   public Student1() {
   }
   public String getRegNo() {
      return regNo;
   }
   public void setRegNo(String regNo) {
      this.regNo = regNo;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public String getCourse() {
      return course;
   }
   public void setCourse(String course) {
      this.course = course;
   }
   public String getSem() {
      return sem;
   }
   public void setSem(String sem) {
      this.sem = sem;
   }
}
```

index.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Student info</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
```

```
<body>
  <h2>Enter Student details</h2>
  <form method="POST" action="Firstpage.jsp">
    <table>
      <tr>
        <td>Register No:</td>
        <td><input type="text" name="regno"></td>
      </tr>
      <tr>
        <td>Name:</td>
        <td><input type="text" name="sname"></td>
      </tr>
      <tr>
        <td>Course:</td>
        <td><input type="text" name="course"></td>
      </tr>
      <tr>
        <td>Semester:</td>
        <td><input type="text" name="sem"></td>
      </tr>
      <tr>
        <td></td>
        <td><input type="submit" name="subBtn" value="Register"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

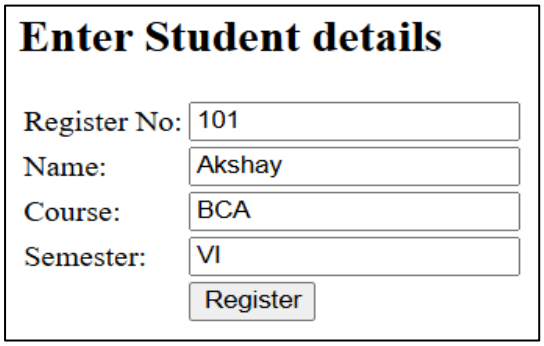
Firstpage.jsp

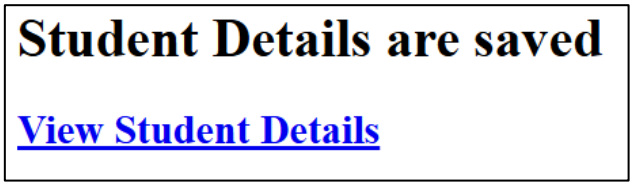
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>First JSP Page</title>
  </head>
  <body>
    <h1>Student Details are saved</h1>
    <jsp:useBean id="std" scope="session" class="com.Student1">
      <jsp:setProperty name="std" property="regNo" value="${param.regno}"/>
      <jsp:setProperty name="std" property="name" value="${param.sname}"/>
      <jsp:setProperty name="std" property="course" value="${param.course}"/>
      <jsp:setProperty name="std" property="sem" value="${param.sem}"/>
    </jsp:useBean>
    <h2><a href="Secondpage.jsp">View Student Details</a></h2>
  </body>
</html>
```

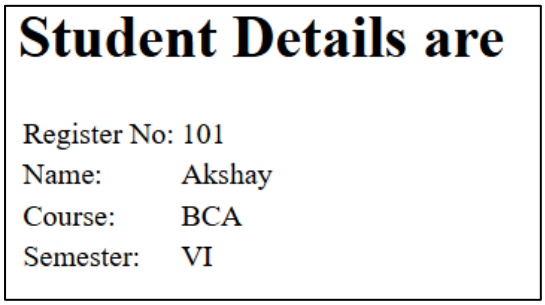
Secondpage.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Second JSP Page</title>
   </head>
   <body>
      <h1>Student Details are</h1>
      <table>
         <tr>
            <td>Register No:</td>
            <td><c:out value="${std.regNo}"/></td>
         </tr>
         <tr>
            <td>Name:</td>
            <td><c:out value="${std.name}"/></td>
         </tr>
         <tr>
            <td>Course:</td>
            <td><c:out value="${std.course}"/></td>
         </tr>
         <tr>
            <td>Semester:</td>
            <td><c:out value="${std.sem}"/></td>
         </tr>
      </table>
   </body>
</html>
```

OUTPUT:







/* ******** */**

PROGRAM: 7

Aim: Write a menu driven program to create a linked list and perform the following operations.

a) To Insert some Elements at the Specified Position
b) Swap two elements in a linked list
c) To Iterate a LinkedList in Reverse Order
d) To Compare Two LinkedList
e) To Convert a LinkedList to ArrayList

Date: 29/04/2025

/* ******** */**

LinkedListDemo.java

```
package linkedlistdemo;
import java.util.*;
public class LinkedListDemo {
   public static void main(String[] args) {
      LinkedList<Integer> flist = new LinkedList<Integer>();
      LinkedList<Integer> slist = new LinkedList<Integer>();
      Scanner in = new Scanner(System.in);
      char choice = 'x';
      int num, pos, fpos, spos;
      do {
         System.out.println("Menu");
         System.out.println("----------------------");
         System.out.println("a. Insert a element");
         System.out.println("b. Swap element");
         System.out.println("c. Iterate in Reverse");
         System.out.println("d. Compare two list");
         System.out.println("e. Convert to Array list");
         System.out.println("x. Exit");
         System.out.println("----------------------");
         System.out.println("Enter your choice->");
         choice = in.next().toLowerCase().charAt(0);
         switch (choice) {
            case 'a':
               if (flist.size() > 0) {
                  System.out.println("Elements in the list:" + flist);
               } else {
                  System.out.println("List is Empty");
               }
               System.out.println("Enter the position:");
               pos = in.nextInt();
               if (pos < 0) {
                  System.out.println("Error! Enter a positive number");
               } else if ((flist.size() > 0 && pos <= flist.size() + 1) || (pos == 1)) {
                  System.out.print("Enter a number:");
                  num = in.nextInt();
                  flist.add(pos - 1, num);
                  System.out.println("Element " + num + " is inserted at " + pos);
               } else {
```

```
            System.out.println("Enter proper position value");
        }
        break;
    case 'b':
        System.out.println("Original List is " + flist);
        System.out.println("Enter the position of the elements to be swapped");
        System.out.println("First element position:");
        fpos = in.nextInt();
        System.out.println("Second element position:");
        spos = in.nextInt();
        if (fpos < 0 && spos <= 0) {
            System.out.println("Error! Use positive value for positions\n");
        } else if ((fpos > flist.size()) && (spos > flist.size())) {
            System.out.println("Error! Enter the positive value for position\n");
        } else {
            int n1 = flist.get(fpos - 1);
            int n2 = flist.get(spos - 1);
            flist.set(spos - 1, n1);
            flist.set(fpos - 1, n2);
            System.out.println("Elements are swapped\n");
            System.out.println("New list is " + flist);
        }
        break;
    case 'c':
        System.out.println("Original list is " + flist);
        System.out.print("Reversed list is [");
        for (Iterator it = flist.descendingIterator(); it.hasNext();) {
            System.out.print(it.next() + " ");
        }
        System.out.println("]\n");
        break;
    case 'd':
        slist = (LinkedList<Integer>) flist.clone();
        Collections.reverse(slist);
        if (flist.equals(slist)) {
            System.out.println("List are equal");
        } else {
            System.out.println("List are not equal");
        }
        break;
    case 'e':
        ArrayList<Integer> alst = new ArrayList<Integer>(flist);
        System.out.println("Element in the array list are");
        System.out.println(alst);
        break;
    case 'x':
        System.out.println("Thank you");
        return;
    default:
        System.out.println("Wrong Choice....\n Try Again!");
```

```
            }
        } while (true);
    }
}
```

OUTPUT:

```
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
a
List is Empty
Enter the position:
1
Enter a number:10
Element 10 is inserted at 1
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
a
Elements in the list:[10]
Enter the position:
2
Enter a number:20
Element 20 is inserted at 2
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
a
Enter the position:
```

```
Elements in the list:[10, 20]
Enter the position:
3
Enter a number:10
Element 10 is inserted at 3
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
c
Original list is [10, 20, 10]
Reversed list is [10 20 10 ]

Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
d
List are equal
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
a
Elements in the list:[10, 20, 10]
4
```

```
Enter a number:30
Element 30 is inserted at 4
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
c
Original list is [10, 20, 10, 30]
Reversed list is [30 10 20 10 ]
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
d
List are not equal
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
b
Original List is [10, 20, 10, 30]
Enter the position of the elements to be
swapped
First element position:
1
```

```
Second element position:
4
Elements are swapped

New list is [30, 20, 10, 10]
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
e
Element in the array list are
[30, 20, 10, 10]
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
f
Wrong Choice....
 Try Again!
Menu
----------------------
a. Insert a element
b. Swap element
c. Iterate in Reverse
d. Compare two list
e. Convert to Array list
x. Exit
----------------------
Enter your choice->
x
Thank you
```

/* ******** */**

PROGRAM: 8

Aim: Implement a java application based on the MVC design pattern.
Input student Rolnlo, name, marks in three subjects calculate result and grade and display the result in neat format.

Date: 29/04/2025

/* ******** */**

StudentModel.java

```
package mvcstudentresult;
public class StudentModel {
   private String rolno, name;
   private int m1, m2, m3;
   public StudentModel(String rolno, String name, int m1, int m2, int m3) {
      this.rolno = rolno;
      this.name = name;
      this.m1 = m1;
      this.m2 = m2;
      this.m3 = m3;
   }
   public String getRolno() {
      return rolno;
   }
   public void setRolno(String rolno) {
      this.rolno = rolno;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public int getM1() {
      return m1;
   }
   public void setM1(int m1) {
      this.m1 = m1;
   }
   public int getM2() {
      return m2;
   }
   public void setM2(int m2) {
      this.m2 = m2;
   }
   public int getM3() {
      return m3;
   }
   public void setM3(int m3) {
      this.m3 = m3;
   }
```

```
    public String getResult() {
        String result = "";
        if (m1 < 35 || m2 < 35 || m3 < 35) {
            result = "Fail";
        } else {
            double per = (((m1 + m2 + m3) * 100) / 300);
            if (per >= 75) {
                result = "Distinction";
            } else if (per >= 60) {
                result = "First Class";
            } else if (per >= 50) {
                result = "Second class";
            } else if (per >= 35) {
                result = "Third class";
            } else {
                result = "Fail";
            }
        }
        return result;
    }
    public String GetGrade() {
        double per = (((m1 + m2 + m3) * 100) / 300);
        String grade = "";
        if (per >= 90) {
            grade = "A";
        } else if (per >= 80) {
            grade = "B";
        } else if (per >= 70) {
            grade = "C";
        } else if (per >= 60) {
            grade = "D";
        } else {
            grade = "E";
        }
        return grade;
    }
}
```

StudentView.java

```
package mvcstudentresult;
public class StudentView {
    public void displayResult(String rNo, String sName, int m1, int m2, int m3, String result,
String grade) {
        System.out.println("_____");
        System.out.println("ROLL NO\t
NAME\t\tMARK1\tMARK2\tMARK3\tRESULT\tGRADE");
        System.out.println(rNo + "\t" + sName + "\t\t" + m1 + "\t" + m2 + "\t" + m3 + "\t" +
result + "\t " + grade);
        System.out.println("_____");
    }
}
```

StudentController.java

```
package mvcstudentresult;
public class StudentController {
   private StudentModel model;
   private StudentView view;
   public StudentController(StudentModel model, StudentView view) {
      this.model = model;
      this.view = view;
   }
   public void UpdateView() {
      view.displayResult(model.getRolno(), model.getName(), model.getM1(),
model.getM2(), model.getM3(), model.getResult(), model.GetGrade());
   }
}
```

MVCStudentResult.java (Run this file to execute)

```
package mvcstudentresult;
import java.util.Scanner;
public class MVCStudentResult {
   public static void main(String[] args) {
      String rNo, sName;
      int m1, m2, m3;
      Scanner in = new Scanner(System.in);
      System.out.print("Enter Roll No:");
      rNo = in.nextLine();
      System.out.print("Enter Name:");
      sName = in.nextLine();
      System.out.print("Marks in three subjects:");
      m1 = in.nextInt();
      m2 = in.nextInt();
      m3 = in.nextInt();
      StudentModel sm = new StudentModel(rNo, sName, m1, m2, m3);
      StudentView sv = new StudentView();
      StudentController sc = new StudentController(sm, sv);
      sc.UpdateView();
   }
}
```

OUTPUT:

```
Enter Roll No:1
Enter Name:Dhanu
Marks in three subjects:21
12
23
```

ROLL NO	NAME	MARK1	MARK2	MARK3	RESULT	GRADE
1	Dhanu	21	12	23	Fail	E