```
1………..import java.util.Scanner;

enum Numwords {
    ZERO("Zero"), ONE("One"), TWO("Two"), THREE("Three"), FOUR("Four"),
    FIVE("Five"), SIX("Six"), SEVEN("Seven"), EIGHT("Eight"), NINE("Nine"),
    TEN("Ten"), ELEVEN("Eleven"), TWELVE("Twelve"), THIRTEEN("Thirteen"),
    FOURTEEN("Fourteen"), FIFTEEN("Fifteen"), SIXTEEN("Sixteen"),
    SEVENTEEN("Seventeen"), EIGHTEEN("Eighteen"), NINETEEN("Nineteen"),
    TWENTY("Twenty"), THIRTY("Thirty"), FORTY("Forty"), FIFTY("Fifty"),
    SIXTY("Sixty"), SEVENTY("Seventy"), EIGHTY("Eighty"), NINETY("Ninety");

    private final String word;

    Numwords(String word) {
        this.word = word;
    }

    public String getWord() {
        return word;
    }
}

public class A1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = sc.nextInt();
        sc.close();

        if (number < 0 || number > 99999) {
```

```
        System.out.println("Number out of range");
    } else {
        String result = convertToWords(number);
        System.out.println("Number in words: " + result);
    }
}

private static String convertToWords(int number) {
    if (number == 0) {
        return Numwords.ZERO.getWord();
    }

    StringBuilder result = new StringBuilder();

    if (number >= 1000) {
        result.append(getWordsForThousands(number / 1000));
        number %= 1000;
    }

    if (number >= 100) {
        result.append(getWordsForHundreds(number / 100));
        number %= 100;
    }

    if (number >= 20) {
        result.append(getWordsForTens(number / 10 * 10));
        number %= 10;
    }

    if (number > 0) {
        result.append(getWordsForUnits(number));
        }

        return result.toString().trim();
    }

    private static String getWordsForThousands(int number) {
        return convertToWords(number) + " Thousand ";
    }

    private static String getWordsForHundreds(int number) {
        return Numwords.values()[number].getWord() + " Hundred ";
    }

    private static String getWordsForTens(int number) {
        return Numwords.values()[number / 10 + 18].getWord() + " ";
    }

    private static String getWordsForUnits(int number) {
        return Numwords.values()[number].getWord() + " ";
    }
}
```

2……

```
package a2;
import java.util.*;
public class A2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        ArrayList<Integer> numbers = new ArrayList<>();
        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            numbers.add(scanner.nextInt()); // Auto-boxing
        }

        Collections.sort(numbers);

        int secondMax = numbers.get(numbers.size() - 2); // Auto-unboxing
        int secondMin = numbers.get(1); // Auto-unboxing

        System.out.println("Second maximum: " + secondMax);
        System.out.println("Second minimum: " + secondMin);
    }
}
```

```
3……….package a3;

import java.util.*;

public class A3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Integer> arrayList = new ArrayList<>();

        int choice;
        do {
            System.out.println("\nMenu:");
            System.out.println("1. Add elements");
            System.out.println("2. Sort elements");
            System.out.println("3. Replace an element");
            System.out.println("4. Remove an element");
            System.out.println("5. Display all elements");
            System.out.println("6. Add an element between two elements");
            System.out.println("7. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1: {
                    System.out.print("Enter number of elements to add: ");
                    int numElements = scanner.nextInt();
                    System.out.println("Enter elements:");
                   for (int i = 0; i < numElements; i++) {
                        int element = scanner.nextInt();
                        arrayList.add(element);
                    }

                    break;
                }
                case 2: {
                    Collections.sort(arrayList);
                    System.out.println("Elements sorted.");
                    break;
                }
                case 3: {
                    System.out.print("Enter index of element to replace: ");
                    int replaceIndex = scanner.nextInt();
                    System.out.print("Enter new element: ");
                    int newElement = scanner.nextInt();
                    arrayList.set(replaceIndex, newElement);
                    System.out.println("Element replaced.");
                    break;
```

```
                }
                case 4: {
                    System.out.print("Enter index of element to remove: ");
                    int removeIndex = scanner.nextInt();
                    arrayList.remove(removeIndex);
                    System.out.println("Element removed.");
                    break;
                }
                case 5: {
                    System.out.println("All elements: " + arrayList);
                    break;
                }
                case 6: {
                    System.out.print("Enter index after which to add element: ");
                    int addIndex = scanner.nextInt();
                    System.out.print("Enter element to add: ");
                    int newElementToAdd = scanner.nextInt();
                    arrayList.add(addIndex + 1, newElementToAdd);
                    System.out.println("Element added.");
                    break;
                }
                case 7:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 7);

        scanner.close();
    }
```

```
}
```

4……..

```
package a4;

import java.util.Scanner;

public class A4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a sentence:");
        String sentence = scanner.nextLine();
        scanner.close();

        System.out.println("Output 1: " + swapEvenWords(sentence));
        System.out.println("Output 2: " + toggleCase(sentence));
    }

       public static String swapEvenWords(String sentence) {
        StringBuilder result = new StringBuilder();
        String[] words = sentence.split("\\s+");

        for (String word : words) {
            if (word.length() % 2 == 0) {
                StringBuilder swapped = new StringBuilder();
                for (int i = 0; i < word.length() - 1; i += 2) {
                    swapped.append(word.charAt(i + 1));
                    swapped.append(word.charAt(i));
                }
                result.append(swapped).append(" ");
            }
        }
        return result.toString().trim();
    }

    public static String toggleCase(String sentence) {
        StringBuilder result = new StringBuilder();
        for (char c : sentence.toCharArray()) {
            if (Character.isUpperCase(c)) {
                result.append(Character.toLowerCase(c));
            } else if (Character.isLowerCase(c)) {
                result.append(Character.toUpperCase(c));
            } else {
                result.append(c);
            }
        }
        return result.toString();
    }
}
```

5………

```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="voting">
    <table style="border:2px solid black; background:lightblue;">
        <tr>
            <td>Name: <input type="text" name="name" placeholder="Enter name" required /></td>
        </tr>
        <tr>
            <td>Age: <input type="number" name="age" placeholder="Enter age" required /></td>
        </tr>
        <tr>
            <td><input type="submit" value="Check Voting Eligibility" /></td>
        </tr>
    </table>
</form>
</body>
</html>
```

```
package age;

import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet("/voting")
public class Voting extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        int age = Integer.parseInt(request.getParameter("age"));

        out.println("<html><body>");
        if (age >= 18) {
          out.println("<h2 style='color:green;'>" + name + ", you are eligible to vote.</h2>");
        } else {
             out.println("<h2 style='color:red;'>" + name + ", you are not eligible to vote.</h2>");
        }
        out.println("<br><a href=\"index.jsp\">Home</a>");
        out.println("</body></html>");
```

```
  }
}
```

6…………<!DOCTYPE html>

```
<html>
<head>
  <title>Prime and Fibonacci Numbers</title>
</head>
<body>
  <h1>First 10 Prime Numbers</h1>
  <%
  int primeCount = 0;
  int number = 2;

  while (primeCount < 10) {
    boolean isPrime = true;
    for (int i = 2; i <= number / 2; i++) {
      if (number % i == 0) {
        isPrime = false;
        break;
      }
    }
    if (isPrime) {
      out.print("<h3>" + number + "</h3>");
      primeCount++;
    }
    number++;
  }
  %>

  <h1>First 10 Numbers of the Fibonacci Series</h1>
  <%
```

```
    int f1 = 0, f2 = 1;
    for (int i = 0; i < 10; i++) {
        out.print("<h3>" + f1 + "</h3>");
        int next = f1 + f2;
        f1 = f2;
        f2 = next;
    }
    %>
</body>
</html>
```

7……………

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
    // Check if the session attribute "cart" exists
    Object cartObj = session.getAttribute("cart");

    // If "cart" exists, check its type
    java.util.HashMap<String, Integer> cart;
    if (cartObj instanceof java.util.HashMap) {
        cart = (java.util.HashMap<String, Integer>) cartObj;
    } else {
        // If "cart" is null or not a HashMap, create a new one and set it in the session
        cart = new java.util.HashMap<String, Integer>();
        session.setAttribute("cart", cart);
    }
%>

<!DOCTYPE html>
<html>
<head>
    <title>Shopping Cart</title>
    <style>
        table {
            width: 50%;
            border-collapse: collapse;
            margin-top: 20px;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>

<h2>Add Item to Cart</h2>
<form action="additem.jsp" method="post">
    <label>Item Name:</label>
    <input type="text" name="item" required>
    <input type="submit" value="Add to Cart">
</form>

<h2>Shopping Cart</h2>
<table>
    <tr>
        <th>Item</th>
        <th>Quantity</th>
        <th>Action</th>
    </tr>
    <%
        for(String item : cart.keySet()) {
    %>
    <tr>
        <td><%= item %></td>
        <td><%= cart.get(item) %></td>
        <td><a href="removeitem.jsp?item=<%= item %>">Remove</a></td>
    </tr>
    <%
        }
    %>
</table>

</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
    String item = request.getParameter("item");
    java.util.HashMap<String, Integer> cart = (java.util.HashMap<String, Integer>)
session.getAttribute("cart");

    if(cart.containsKey(item)) {
        cart.put(item, cart.get(item) + 1);
    } else {
        cart.put(item, 1);
    }

    response.sendRedirect("index.jsp");
```

```
%>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
    String item = request.getParameter("item");
    java.util.HashMap<String, Integer> cart = (java.util.HashMap<String, Integer>)
session.getAttribute("cart");

    if(cart.containsKey(item)) {
        if(cart.get(item) > 1) {
            cart.put(item, cart.get(item) - 1);
        } else {
            cart.remove(item);
        }
    }

    response.sendRedirect("index.jsp");
%>
```

8……….

```
<html>
    <body>
        <a href="fileservlet">Download the jsp file</a>
        <br><br>
    </body>
</html>
```

```
package filepck;
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 fileservlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out=response.getWriter();
        String filename="E:\shetty\testjava.java";
        String filepath="E:\shetty\testjava.java";
        response.setContentType("APPLICATION/OCTET-STREAM");
        response.setHeader("content-disposition","attachment; filename=\"testjava.java\"");
        java.io.FileInputStream fs = new java.io.FileInputStream("E:/shetty/testjava.java");
        int i;
        while((i=fs.read())!=-1){
            out.write(i);
        }
        fs.close();
        out.close();
    }
}
```

B1…………….

```
package studentdatabase;

import java.sql.*;
import java.util.Scanner;

public class StudentDatabase {
    private static final String DB_URL = 
"jdbc:mysql://localhost:3306/sddata?characterEncoding=latin1&useConfigs=maxPerformance";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static void main(String[] args) {
        try
            {
            Scanner sc = new Scanner(System.in);
            Connection connection = DriverManager.getConnection(DB_URL, USER, PASSWORD);
           int choice;
           do {
               System.out.println("\nMENU:");
               System.out.println("1. Add New Record");
               System.out.println("2. Update Address");
               System.out.println("3. Delete Record");
               System.out.println("4. Search for a Student");
               System.out.println("5. Exit");
               System.out.print("Enter your choice: ");
               choice = sc.nextInt();
               sc.nextLine();  // Consume newline

               switch (choice) {
                   case 1:
                       addRecord(connection, sc);
                       break;
                   case 2:
                       updateStudentAddress(connection, sc);
                       break;
                   case 3:
                       deleteRecord(connection, sc);
```

```
                    break;
                case 4:
                    searchStudent(connection, sc);
                    break;
                case 5:
                    System.out.println("Exiting Program");
                    break;

            }
        } while (choice != 5);
    } catch (SQLException e) {
        System.err.println("Error: " + e.getMessage());
    }
}

public static void addRecord(Connection connection, Scanner sc) {
    System.out.print("Enter Student RegNo: ");
    int regNo = sc.nextInt();
    sc.nextLine();  // Consume newline
    System.out.print("Enter Student Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Student DoB (YYYY-MM-DD): ");
    String dob = sc.nextLine();
    System.out.print("Enter Student Address: ");
    String address = sc.nextLine();
    System.out.print("Enter Student Class: ");
    String studentClass = sc.nextLine();
    System.out.print("Enter Student Course: ");
    String course = sc.nextLine();
```

```
        String query = "INSERT INTO student (regno, name, dob, address, class, course) VALUES (?, ?, ?,
?, ?, ?)";
        try  {
            PreparedStatement statement = connection.prepareStatement(query);

            statement.setInt(1, regNo);
            statement.setString(2, name);
            statement.setString(3, dob);
            statement.setString(4, address);
            statement.setString(5, studentClass);
            statement.setString(6, course);
            statement.executeUpdate();
            System.out.println("Record Added Successfully");
        } catch (SQLException e) {
            System.err.println("Error adding record: " + e.getMessage());
        }
    }

    public static void updateStudentAddress(Connection connection, Scanner sc) {
        System.out.print("Enter Student RegNo to update address: ");
        int regNo = sc.nextInt();
        sc.nextLine();  // Consume newline
        System.out.print("Enter new address: ");
        String address = sc.nextLine();

        String query = "UPDATE student SET address = ? WHERE regno = ?";
        try  {
                        PreparedStatement statement = connection.prepareStatement(query);

            statement.setString(1, address);
            statement.setInt(2, regNo);
        int rowsUpdated = statement.executeUpdate();
        if (rowsUpdated > 0) {
            System.out.println("Student address updated successfully");
        } else {
            System.out.println("No student found with RegNo: " + regNo);
        }
    } catch (SQLException e) {
        System.err.println("Error updating address: " + e.getMessage());
    }
}

public static void deleteRecord(Connection connection, Scanner sc) {
    System.out.print("Enter Student RegNo to delete: ");
    int regNo = sc.nextInt();
    sc.nextLine();  // Consume newline

    String query = "DELETE FROM student WHERE regno = ?";
    try  {
                    PreparedStatement statement = connection.prepareStatement(query);

        statement.setInt(1, regNo);
        int rowsDeleted = statement.executeUpdate();
        if (rowsDeleted > 0) {
            System.out.println("Student record deleted successfully");
        } else {
            System.out.println("No student found with RegNo: " + regNo);
        }
    } catch (SQLException e) {
        System.err.println("Error deleting record: " + e.getMessage());
    }
}
```

```
    public static void searchStudent(Connection connection, Scanner sc) {
        System.out.print("Enter Student RegNo to search: ");
        int regNo = sc.nextInt();
        sc.nextLine();  // Consume newline

        String query = "SELECT * FROM student WHERE regno = ?";
        try  {
      PreparedStatement statement = connection.prepareStatement(query);

            statement.setInt(1, regNo);

                ResultSet resultSet = statement.executeQuery();
                if (resultSet.next()) {
                    String name = resultSet.getString("name");
                    String dob = resultSet.getString("dob");
                    String address = resultSet.getString("address");
                    String studentClass = resultSet.getString("class");
                    String course = resultSet.getString("course");
                    System.out.printf("RegNo: %d, Name: %s, DoB: %s, Address: %s, Class: %s, Course: 
%s%n",
                            regNo, name, dob, address, studentClass, course);
                } else {
                    System.out.println("No student found with RegNo: " + regNo);
                }

        } catch (SQLException e) {
            System.err.println("Error searching for student: " + e.getMessage());
        }
    }
}
```

B2……….

```
package bank;


import java.sql.*;
import java.util.Scanner;


public class Bank {
//    private static final String DB_URL = "jdbc:mysql://localhost/bank";
     private static final String DB_URL =
"jdbc:mysql://localhost:3306/bank?characterEncoding=latin1&useConfigs=maxPerformance";
    private static final String USER = "root";
    private static final String PASSWORD = "";


    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
             Connection connection = DriverManager.getConnection(DB_URL, USER, PASSWORD);
            int choice;
            do {
                System.out.println("\nMENU:");
                System.out.println("1. Add New Account");
                System.out.println("2. Deposit");
                System.out.println("3. Withdraw");
                System.out.println("4. Display Accounts");
                System.out.println("5. Exit");
                System.out.print("Enter your choice: ");
                choice = sc.nextInt();
                switch (choice) {
                    case 1:
                        addAccount(connection, sc);
                        break;
                case 2:
                    deposit(connection, sc);
                    break;
                case 3:
                    withdraw(connection, sc);
                    break;
                case 4:

                    display(connection);
                    break;
                case 5:
                    System.out.println("Exiting Program");
                    break;
                default:
                    System.out.println("Invalid Choice!");
            }
        } while (choice != 5);
    } catch (SQLException e) {
        System.err.println("Error: " + e.getMessage());
    }
}

public static void addAccount(Connection connection, Scanner sc) {
    System.out.print("Enter account number: ");
    int accno = sc.nextInt();
    System.out.print("Enter account holder name: ");
    String name = sc.next();
    System.out.print("Enter initial balance: ");
    int balance = sc.nextInt();

    String query = "SELECT * FROM bankdb WHERE accno = ?";
```

```
        try  {
          PreparedStatement checkStmt = connection.prepareStatement(query);
          checkStmt.setInt(1, accno);
          ResultSet rs = checkStmt.executeQuery();
          if (rs.next()) {
            System.out.println("Account already exists.");
            return;
          }

        String insertQuery = "INSERT INTO bankdb (accno, accname, bal) VALUES (?, ?, ?)";
          PreparedStatement stmt = connection.prepareStatement(insertQuery);
          stmt.setInt(1, accno);
          stmt.setString(2, name);
          stmt.setInt(3, balance);
          stmt.executeUpdate();
          System.out.println("Account added successfully.");
        }catch (SQLException e) {
          System.err.println("Error adding account: " + e.getMessage());
        }
      }


    public static void display(Connection connection) {
      String query = "SELECT * FROM bankdb";
      try {
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("\nRECORDS:");
        while (rs.next()) {
          int accno = rs.getInt("accno");
          String name = rs.getString("accname");
```

```
            int balance = rs.getInt("bal");
            System.out.printf("Accno: %d, Name: %s, Balance: %d%n", accno, name, balance);
        }
    } catch (SQLException e) {
        System.err.println("Error displaying accounts: " + e.getMessage());
    }
}


    public static void deposit(Connection connection, Scanner sc) {
        System.out.print("Enter account number: ");
        int accno = sc.nextInt();

        String checkQuery = "SELECT 1 FROM bankdb WHERE accno = ?";
        try  {
            PreparedStatement checkStmt = connection.prepareStatement(checkQuery);
            checkStmt.setInt(1, accno);
            ResultSet rs = checkStmt.executeQuery();
            if (!rs.next()) {
                System.out.println("No such account.");
                return;
            }



        System.out.print("Enter amount to deposit: ");
        int amount = sc.nextInt();

        String updateQuery = "UPDATE bankdb SET bal = bal + ? WHERE accno = ?";

            PreparedStatement stmt = connection.prepareStatement(updateQuery);
            stmt.setInt(1, amount);
```

```
      stmt.setInt(2, accno);
      stmt.executeUpdate();
      System.out.println("Amount deposited successfully.");
   } catch (SQLException e) {
      System.err.println("Error depositing amount: " + e.getMessage());
   }
}

public static void withdraw(Connection connection, Scanner sc) {
   System.out.print("Enter account number: ");
   int accno = sc.nextInt();

   String selectQuery = "SELECT bal FROM bankdb WHERE accno = ?";
   try {
       PreparedStatement selectStmt = connection.prepareStatement(selectQuery);
      selectStmt.setInt(1, accno);
      ResultSet rs = selectStmt.executeQuery();
      if (!rs.next()) {
         System.out.println("No such account.");
         return;
      }

      int currentBalance = rs.getInt("bal");
      System.out.print("Enter amount to withdraw: ");
      int amount = sc.nextInt();

      if (amount <= 0) {
         System.out.println("Withdrawal amount should be greater than 0.");
         return;
      }
      int remainingBalance = currentBalance - amount;
      if (remainingBalance < 500) {
        System.out.println("Minimum balance must be maintained. Current balance is " +
currentBalance);
        return;
      }

      String updateQuery = "UPDATE bankdb SET bal = bal - ? WHERE accno = ?";

        PreparedStatement updateStmt = connection.prepareStatement(updateQuery);
        updateStmt.setInt(1, amount);
        updateStmt.setInt(2, accno);
        updateStmt.executeUpdate();
        System.out.println("Amount withdrawn successfully.");

    } catch (SQLException e) {
      System.err.println("Error checking account balance: " + e.getMessage());
    }
  }
}
```

B3:

Servent.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface Servent extends Remote
{
 public double calculateTax(double amount)throws RemoteException;
}
```

Tax.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
```

```
public class Tax extends UnicastRemoteObject implements Servent
{
 public Tax() throws RemoteException
 {
  super();
 }

 @Override
 public double calculateTax(double amount) throws RemoteException
 {
  if(amount<=300000)
   return 0;
  else if(amount<=600000)
   return (0.05*amount);
  else if(amount<=900000)
   return (0.1*amount);
  else if(amount<=1200000)
   return (0.15*amount);
  else if(amount<=1500000)
   return (0.2*amount);
  else
   return (0.3*amount);
 }
}
```

Server.java

```
import java.rmi.*;

public class Server
{
 public static void main(String[] args)throws Exception
 {
  Tax obj=new Tax();
  Naming.rebind("TAX",obj);
  System.out.println("Server Ready...");
 }
}
```

TaxClient.java

```
import java.rmi.*;
public class TaxClient
{
 public static void main(String[] args)throws Exception
 {
  if(args.length!=2)
  {
   System.out.println("Irrelevent number of arguments...");
   System.exit(1);
  }

  try
  {
   Servent obj=(Servent)Naming.lookup("TAX");

   String name=(args[0]);
   double amount=Double.parseDouble(args[1]);
   double tax=obj.calculateTax(amount);
   System.out.println("Tax of "+name+ " is: "+tax);
  }
  catch (Exception e)
  {
   System.out.println("Exception");
  }
 }
}
```

B4:

SimpleInterestInterface.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface SimpleInterestInterface extends Remote
{
  public double calculateSimpleInterest(double principle,double rate,double time)throws
RemoteException;
}
```

SimpleInterest.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class SimpleInterest extends UnicastRemoteObject implements SimpleInterestInterface
{
  public SimpleInterest() throws RemoteException
  {
   super();
  }

  @Override
  public double calculateSimpleInterest(double principle,double rate,double time) throws
RemoteException
 {
  return (principle*rate*time)/100;
 }
}
```

Server.java

```
import java.rmi.*;
```

```
public class Server
{
 public static void main(String[] args)throws Exception
 {
  SimpleInterest obj=new SimpleInterest();
  Naming.rebind("SIMPLE",obj);
  System.out.println("Server Ready...");
 }
}
```

SimpleInterestClient.java

```
import java.rmi.*;
public class SimpleInterestClient
{
 public static void main(String[] args)throws Exception
 {
  if(args.length!=3)
  {
   System.out.println("Irrelevent number of arguments...");
   System.exit(1);
  }
  try
  {

   SimpleInterestInterface obj=(SimpleInterestInterface)Naming.lookup("SIMPLE");

   double principle=Double.parseDouble(args[0]);
   double rate=Double.parseDouble(args[1]);
   double time=Double.parseDouble(args[2]);
   double interest=obj.calculateSimpleInterest(principle,rate,time);
   System.out.println("Simple Interest: "+interest);
  }
  catch (Exception e)
  {
   System.out.println("Exception");
  }
 }
}
```

B6……..

```
<%--
    Document   : studentgui
    Created on : Apr 4, 2024, 9:24:21 AM
    Author     : mr
--%>
<%@page import="pckindex.student"%>
<%@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>Enter Student Details</h1>
        <form action="displaystudent.jsp" method="post">
            Name:<input type="text" name="name"></input>
            RollNo:<input type="number" name="rollno"></input>
            Class:<input type="text" name="cls">   </input>
            Section:<input type="text" name="sec"> </input>
            <input type="submit" value="Submit"> </input>
        </form>
```

```
  </body>
</html>


<%@page import="pckindex.student"%>
<%@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>Student Details</h1>
    <%
      String name=request.getParameter("name");
      int rollno=Integer.parseInt(request.getParameter("rollno"));
      String cls=request.getParameter("cls");
      String sec=request.getParameter("sec");
      student stu=new student();
      stu.setName(name);
      stu.setRollNo(rollno);
      stu.setCls(cls);
      stu.setSec(sec);


      %>
      <p>Name:<%=stu.getName()%></p>
      <p>Roll No:<%=stu.getRollNO()%></p>
      <p>Class:<%=stu.getCls()%></p>
      <p>Roll No:<%=stu.getSec()%></p>
  </body>
</html>
```

```
package pckindex;

public class student {
    private String name;
    private int rollno;
    private String cls;
    private String sec;
    public student(){

    }
    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name=name;
    }
    public int getRollNO(){
        return rollno;
    }
    public void setRollNo(int rollno){
        this.rollno=rollno;
    }
    public String getCls(){
        return cls;
    }
    public void setCls(String cls) {
        this.cls=cls;
    }
    public String getSec(){
        return sec;
    }
    public void setSec(String sec){
        this.sec=sec;
    }
    }
```

B7………….

```
package linkedlistoperation;

import java.util.*;

public class Linkedlistoperation {

    public static void main(String[] args) {
        // TODO code application logic here
        Scanner sc=new Scanner(System.in);
        LinkedList<Integer> linkedlist=new LinkedList<Integer>();
        System.out.println("Enter the number of elements to create linkedlist:");
        int n=sc.nextInt();
        System.out.println("Enter the elements: ");

        for(int i=0;i<n;i++)
        {
          int e=sc.nextInt();
          linkedlist.add(e);
        }
        while(true)
        {
            System.out.println("1.To insert");
            System.out.println("2.To Swap");
            System.out.println("3.To iterate reverse order");
            System.out.println("4.Compare");
            System.out.println("5.Convert linked list to arraylist");
            System.out.println("6.Exit");
            System.out.println("Enter choice");
            int choice=sc.nextInt();
            switch(choice){
              case 1:
                System.out.print("enter position");
                int pos=sc.nextInt();
                System.out.print("enter element");
                int ele=sc.nextInt();
                linkedlist.add(pos,ele);
                System.out.println("LinkedList is"+linkedlist);
                     break;
              case 2:
                System.out.print("enter index 1 to swap");
                int index1=sc.nextInt();
                System.out.print("enter index 2");
                int index2=sc.nextInt();
                Collections.swap(linkedlist,index1,index2);
                System.out.println("LinkedList is"+linkedlist);
                break;
              case 3:
                System.out.println("reverse order");
                ListIterator<Integer> iterator=linkedlist.listIterator(linkedlist.size());
                while(iterator.hasPrevious())
                {
                  System.out.println(iterator.previous()+"");

                }
                System.out.println();
                break;
```

```
                case 4:
                    LinkedList<Integer> secondlist=new LinkedList<Integer>();
                    System.out.print("Enter elements for the linked list(seperated by space)");
                    sc.nextLine();
                    String[] elements=sc.nextLine().split(" ");
                    for(String eles:elements)
                    {
                        secondlist.add(Integer.parseInt(eles));

                    }
                    System.out.println("Comparison result"+linkedlist.equals(secondlist));
                    break;
                case 5:
                    ArrayList<Integer> arraylist=new ArrayList<Integer>(linkedlist);
                    System.out.println("Linkedlist converted to arraylist"+arraylist);
                    break;
                case 6:
                    System.out.println("Exiting program ");
                    System.exit(0);
            }
        }

    }
}
```

B8…………

```
package student;

import java.util.Scanner;

    class studentcontoller
    {
```

```
private Student model;
private studentview view;


public  studentcontoller(Student model,studentview view)
{
   this.model=model;
   this.view=view;
}


public void setStudentName(String StudentName)
{
   model.setStudentName(StudentName);
}
public String getStudentName()
{
   return model.getStudentName();
}

public void setStudentRollno(String StudentRollno)
{
   model.setStudentRollno(StudentRollno);
}
public String getStudentRollno()
{
   return model.getStudentRollno();
}
public void setMark1(Integer mark1)
{
   model.setMark1(mark1);
   }
   public Integer getMark1()
   {
      return model.getMark1();
   }
   public void setMark2(Integer mark2)
   {
      model.setMark2(mark2);
   }
   public Integer getMark2()
   {
      return model.getMark2();
   }
    public void setMark3(Integer mark3)
   {
      model.setMark3(mark3);
   }
   public Integer getMark3()
   {
      return model.getMark3();
   }
   public void updateView(){
      view.printStddetails(model.getStudentName(),model.getStudentRollno(),model.getMark1(),
model.getMark2(),model.getMark3());
   }


 }
public class stdMain{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the no.of students:");
        int n = sc.nextInt();
        for(int i=0;i<n;i++)
        {
        studentview view = new studentview();
        Student model = AddInfo();
        studentcontoller controller = new  studentcontoller(model,view);
        controller.updateView();
        }

    }
    private static Student AddInfo()
    {
        Student std = new Student();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Name:");
        std.setStudentName(sc.nextLine());
        System.out.println("Enter the Roll No:");
        std.setStudentRollno(sc.nextLine());
        System.out.println("Enter the Mark1:");
        std.setMark1(sc.nextInt());
        System.out.println("Enter the Mark2:");
        std.setMark2(sc.nextInt());
        System.out.println("Enter the Mark3:");
        std.setMark3(sc.nextInt());
        return std;
    }
}

package student;
public class Student {
```

```
private String StudentName;
private String StudentRollno;
private Integer Mark1;
private Integer Mark2;
private Integer Mark3;


public String getStudentName() {
   return StudentName;
}

public void setStudentName(String StudentName) {
   this.StudentName = StudentName;
}

public String getStudentRollno() {
   return StudentRollno;
}

public void setStudentRollno(String StudentRollno) {
   this.StudentRollno = StudentRollno;
}

public Integer getMark1() {
   return Mark1;
}

public void setMark1(Integer Mark1) {
   this.Mark1 = Mark1;
}
```

```
    public Integer getMark2() {
        return Mark2;
    }

    public void setMark2(Integer Mark2) {
        this.Mark2 = Mark2;
    }

    public Integer getMark3() {
        return Mark3;
    }

    public void setMark3(Integer Mark3) {
        this.Mark3 = Mark3;
    }
}
package  student;
public class studentview {
    public void printStddetails(String StudentName,String StudentRollno,Integer Mark1,
            Integer Mark2,Integer Mark3)
    {
        System.out.println("Student Details");
        System.out.println("Name:"+StudentName);
        System.out.println("Student Rollno:"+StudentRollno);
        System.out.println("Subject Mark1:"+Mark1);
        System.out.println("Subject Mark2:"+Mark2);
        System.out.println("Subject Mark3:"+Mark3);
        int total = Mark1+Mark2+Mark3;
        float percentage = total/3;
        String grade;
        if(Mark1<35 || Mark2 <35 || Mark3 < 35){
            grade="Fail";

        }else{
        if(percentage >90){
            grade="A";
        }else if(percentage<=90 && percentage >80){
            grade="B";
        }else if(percentage<=80 && percentage >70){
            grade="C";
        }else if(percentage<=70 && percentage >60){
            grade="D";
        }else{
            grade="E";
        }
        }

        System.out.println("Total:"+total);
        System.out.println("Grade:"+grade);
    }
}
```