----------**1)AirVoiceMain.java**----------

```
import java.util.*;

public class Main {


	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);


		String name,email;

		int age;

		long number;


		System.out.println("Enter the Name");

		name =sc.nextLine();

		System.out.println("Enter the Contact Number");

		number =sc.nextLong();

		System.out.println("Enter the Emailid");

		String ss=sc.nextLine();

		email =sc.nextLine();

		System.out.println("Enter the Age:");

		age =sc.nextInt();


		System.out.println("");


		Customer c = new Customer();


		c.setCustomerName(name);
```

```
		c.setContactNumber(number);
		c.setEmailId(email);
		c.setAge(age);

		String mail = c.getEmailId();
		System.out.println("Name:"+c.getCustomerName());
		System.out.println("ContactNumber:"+c.getContactNumber());
		System.out.println("EmailId:"+mail);
		System.out.println("Name:"+c.getAge());
		sc.close();
	}
}
```

----------Cutomer.java----------

```
public class Customer {
	private String customerName;
	private long contactNumber;
	private String emailId;
	private int age;
	public String getCustomerName() {
		return customerName;
	}
	public void setCustomerName(String customerName) {
		this.customerName = customerName;
	}
```

```
	public long getContactNumber() {
		return contactNumber;
	}
	public void setContactNumber(long contactNumber) {
		this.contactNumber = contactNumber;
	}
	public String getEmailId() {
		return emailId;
	}
	public void setEmailId(String emailId) {
		this.emailId = emailId;
	}
	public int getAge() {
		return age;
	}
	public void setAge(int age) {
		this.age = age;
	}


}
```

2) Call Details

-------CallMain.java-------

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;


public class CallMain {



   public static void main(String args[]) throws IOException


   {


      BufferedReader br=new BufferedReader( new InputStreamReader( System.in ) );


      String cd=br.readLine();


      Call c=new Call( );


      c.parseData(cd);


      System.out.println("CalId :"+c.getCallId());


      System.out.println( "Called Number :"+c.getCalledNumber() );


      System.out.println("Duration :"+c.getDuration() );
```

```
    }

}
```

--------Call.java-------

```
public class Call {
        private int callId;

    private long calledNumber;

    private float duration;



    public void parseData(String cd) {

        String callDetails[] = cd.split( ":" );

        this.callId = Integer.parseInt( callDetails[0] );

        this.calledNumber = Long.parseLong( callDetails[1] );

        this.duration = Float.parseFloat( callDetails[2] );
```

```
}

public void setCallId(int callId) {
    this.callId = callId;
}

public void setCalledNumber(long calledNumber) {
    this.calledNumber = calledNumber;
}

public void setDuration(float duration) {
    this.duration = duration;
```

```
}

public float getDuration() {
    return duration;
}

public int getCallId() {
    return callId;
}

public long getCalledNumber() {
    return calledNumber;
}
```

```
}
```

3) Pair of Two digits

```
-------Paire of 2 Digits------
import java.util.*;
public class Product_of_number {
          static int check(int num)
   {

      int reversenum = 0;


      while(num > 0)
      {
         int rem = num % 10;
         reversenum = reversenum * 10
                     + rem;
         num = num / 10;
      }
     return reversenum;


   }


         public static void main(String[] args) {
                  Scanner sc =new Scanner(System.in);
                  System.out.println("Enter num1 and num 2");
```

```
                int num1 = sc.nextInt();
                int num2 = sc.nextInt();
        // call check() function.
            int product1 = num1*num2;
            int product2 = check(num1)*check(num2);

        if (product1 == product2)
           System.out.printf("%d and %d are correct pair.",num1,num2);
        else
          System.out.printf("%d and %d are not correct pair.",num1,num2);
            sc.close();
    }

}
```

4) ZeeZee bank

```
----------ZeeZeeBankMain----------
import java.text.DecimalFormat;
import java.util.Scanner;
public class Main {
        public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);
                DecimalFormat decimalFormat = new DecimalFormat("0.00");
                System.out.println("Enter the account number:");
```

```
                long accountNumber = scanner.nextLong();

                System.out.println("Enter initial balance:");

                double balanceAmount = scanner.nextDouble();

                Account account = new Account(accountNumber, balanceAmount);

                System.out.println("Enter the amount to be deposited:");

                double depositAmount = scanner.nextDouble();

                account.deposit(depositAmount);

                double availableBalance = account.getBalanceAmount();

                System.out.println("Available balance is:" +
decimalFormat.format(availableBalance));

                System.out.println("Enter the amount to be withdrawn:");

                double withdrawAmount = scanner.nextDouble();

                boolean isWithdrawn = account.withdraw(withdrawAmount);

                availableBalance = account.getBalanceAmount();

                if (!isWithdrawn) {

                System.out.println("Insufficient balance");

                }

                System.out.println("Available balance is:" +
decimalFormat.format(availableBalance));

        scanner.close();

        }

}
```

----------Account.java----------

```
public class Account {
```

```
private long accountNumber;
private double balanceAmount;
public Account(long accountNumber, double balanceAmount) {
this.accountNumber = accountNumber;
this.balanceAmount = balanceAmount;
}
public long getAccountNumber() {
return accountNumber;
}
public void setAccountNumber(long accountNumber) {
this.accountNumber = accountNumber;
}
public double getBalanceAmount() {
return balanceAmount;
}
public void setBalanceAmount(double balanceAmount) {
this.balanceAmount = balanceAmount;
}
public void deposit(double depositAmount) {
balanceAmount += depositAmount;
}
public boolean withdraw(double withdrawAmount) {
if (withdrawAmount <= balanceAmount) {
balanceAmount -= withdrawAmount;
return true;
```

```
	}
	return false;
	}
}
```

5) Find MemberShip Category Count

----------MemberMain.java----------

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class Main {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		List<Member> mList=new ArrayList<Member>();
		System.out.println("Enter the number of Members:");
		Scanner sc=new Scanner(System.in);
		int tot=sc.nextInt();
		String[] str=new String[tot];
		for(int i=0;i<str.length;i++)
		{
		System.out.println("Enter the Member Details:");
		str[i]=sc.next();
		}
```

```
Member m[]=new Member[tot];
for(int i=0;i<m.length;i++)
{
String s[]=str[i].split(":");
m[i]=new Member(s[0],s[1],s[2]);
mList.add(m[i]);
}
System.out.println("Enter the number of times Membership category needs to be searched:");
int tot1=sc.nextInt();
ZeeShop t1[]=new ZeeShop[tot1];
for(int i=0;i<tot1;i++)
{
System.out.println("Enter the Category");
String s1=sc.next();
t1[i]=new ZeeShop(s1,mList);
t1[i].start();
//System.out.println(s1+" "+t1.getCount());
}
try {
Thread.currentThread().sleep(2000);
} catch (InterruptedException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
for(ZeeShop s:t1)
```

```
                {
                System.out.println(s.getMemberCategory()+":"+s.getCount());
                }

        }

}
```

----------Member.java----------

```
public class Member {
        private String memberId;
        private String memberName;
        private String category;
        public String getMemberId() {
        return memberId;
        }
        public void setMemberId(String memberId) {
        this.memberId = memberId;
        }
        public String getMemberName() {
        return memberName;
        }
        public void setMemberName(String memberName) {
        this.memberName = memberName;
```

```
	}
	public String getCategory() {
	return category;
	}
	public void setCategory(String category) {
	this.category = category;
	}
	public Member(String memberId, String memberName, String category) {
	super();
	this.memberId = memberId;
	this.memberName = memberName;
	this.category = category;
	}
}
```

----------ZeeShop.java----------

```
import java.util.List;
public class ZeeShop extends Thread {
	private String memberCategory;
	private int count;
	private List<Member> memberList;
	public ZeeShop(String memberCategory, List<Member> memberList) {
	super();
	this.memberCategory = memberCategory;
```

```
this.memberList = memberList;
}
public String getMemberCategory() {
return memberCategory;
}
public void setMemberCategory(String memberCategory) {
this.memberCategory = memberCategory;
}
public int getCount() {
        return count;
        }
        public void setCount(int count) {
        this.count = count;
        }
        public List<Member> getMemberList() {
        return memberList;
        }
        public void setMemberList(List<Member> memberList) {
        this.memberList = memberList;
        }
        public void run()
        {
        synchronized(this)
        {
        for(Member m:memberList)
```

```
                {
                if(m.getCategory().equals(memberCategory))
                count++;
                }
                }
        }
}
```

6) Grade Calculation

----------GradeCalculationMain.java----------

```
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Main {

        public static void main(String[] args) throws Exception{
                // TODO Auto-generated method stub
                BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
                System.out.println("Enter the number of Threads:");
                int th = Integer.parseInt(br.readLine());
                GradeCalculator obj = null;
                String str = "";
                String details[] = new String[th];
```

```
            for(int i=0; i<th; i++){
            System.out.println("Enter the String:");
            str = br.readLine();
            details[i]=str;
            }

            for(int i=0; i<th; i++){
            String sp[] = details[i].split(":");
            int k = 0;
            int arr[] = new int[sp.length];
            for(int j = 1; j<sp.length; j++)
            arr[k++] = Integer.parseInt(sp[j]);
            obj = new GradeCalculator(sp[0],arr);
            obj.start();
            try{
            Thread.sleep(1000);
            }
            catch(Exception e)
            {
            System.out.println(e);
            }
            }

}
```

```
}
```

----------GradeCalculator.java----------

```
public class GradeCalculator extends Thread{
        private String studName;
         private char result;
         private int[] marks;

         public String getStudName(){
         return studName;
         }

         public void setStudName(String studName){
         this.studName = studName;
         }

         public char getResult(){
         return result;
         }

         public void setResult(char result){
         this.result = result;
```

```
}

public int[] getMarks(){
return marks;
}

public void setMarks(int[] marks){
this.marks = marks;
}
public GradeCalculator(String studName, int[] marks){
        this.studName = studName;
        this.marks = marks;
        }

        public void run(){
        int sum = 0;
        int[] score = getMarks();
        for(int i = 0;i<score.length;i++)
        sum = sum+score[i];
        if((400<=sum)&&(sum<=500))
        System.out.println(getStudName()+":"+'A');
        if((300<=sum)&&(sum<=399))
        System.out.println(getStudName()+":"+'B');
        if((200<=sum)&&(sum<=299))
        System.out.println(getStudName()+":"+'C');
```

```
                    if(sum<200)
                    System.out.println(getStudName()+":"+'E');
                    }
}
```

7) Query Data Set

----------QueryClass.java----------

```
public class Query {
        public class DataSet {
                private String theatreId;
                private String theatreName;
                private String location;
                private int noOfScreen;
                private double ticketCost;
                public String getTheatreId() {
                return theatreId;
                }
                public void setTheatreId(String theatreId) {
                this.theatreId = theatreId;
                }
                public String getTheatreName() {
                return theatreName;
                }
```

```
public void setTheatreName(String theatreName) {
this.theatreName = theatreName;
}
public String getLocation() {
return location;
}
public void setLocation(String location) {
	this.location = location;
	}
	public int getNoOfScreen() {
	return noOfScreen;
	}
	public void setNoOfScreen(int noOfScreen) {
	this.noOfScreen = noOfScreen;
	}
	public double getTicketCost() {
	return ticketCost;
	}
	public void setTicketCost(double ticketCost) {
	this.ticketCost = ticketCost;
	}
	@Override
	public String toString() {
	return "Theatre id: " + theatreId + "\nTheatre name: " + theatreName
+ "\nLocation: " + location
```

```
                        + "\nNo of Screen: " + noOfScreen + "\nTicket Cost: " +
ticketCost+"\n";
                        }
        }
        private String queryId;
        private String queryCategory;
        private DataSet primaryDataset;
        private DataSet secondaryDataSet;
        public String getQueryId() {
        return queryId;
        }
        public void setQueryId(String queryId) {
        this.queryId = queryId;
        }
        public String getQueryCategory() {
        return queryCategory;
        }
        public void setQueryCategory(String queryCategory) {
        this.queryCategory = queryCategory;
        }
        public DataSet getPrimaryDataset() {
        return primaryDataset;
        }
        public void setPrimaryDataset(DataSet primaryDataset) {
        this.primaryDataset = primaryDataset;
        }
```

```
	public DataSet getSecondaryDataSet() {
		return secondaryDataSet;
		}
		public void setSecondaryDataSet(DataSet secondaryDataSet) {
		this.secondaryDataSet = secondaryDataSet;
		}
		@Override
		public String toString() {
		return "Primary data set\n" + primaryDataset
		+ "Secondary data set\n" + secondaryDataSet +"Query id: "+ queryId +
"\nQuery category=" +
		queryCategory;
		}

}
```

----------testApplication.java----------

```
import java.util.*;
public class TestApplication {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Query query = new Query();
		Scanner sc = new Scanner(System.in);
		Query.DataSet primary = query.new DataSet();
```

```
Query.DataSet secondary = query.new DataSet();
System.out.println("Enter the Details of primary data set");
System.out.println("Enter the theatre id");
String theatreid = sc.next();
primary.setTheatreId(theatreid);
sc.nextLine();
System.out.println("Enter the theatre name");
String theatrename = sc.next();
primary.setTheatreName(theatrename);
sc.nextLine();
System.out.println("Enter the location");
String location = sc.next();
primary.setLocation(location);
sc.nextLine();
System.out.println("Entrer the no of screens");
int screens = sc.nextInt();
primary.setNoOfScreen(screens);
System.out.println("Ente the ticket cost");
double cost = sc.nextDouble();
primary.setTicketCost(cost);
System.out.println("ENter the details of secondary data set");
System.out.println("Enter the theatre id");
theatreid = sc.next();
secondary.setTheatreId(theatreid);
sc.nextLine();
```

```
System.out.println("Enter the theatre name");
theatrename = sc.next();
secondary.setTheatreName(theatrename);
sc.nextLine();
System.out.println("Enter the location");
location = sc.next();
secondary.setLocation(location);
sc.nextLine();
System.out.println("Entrer the no of screens");
screens = sc.nextInt();
secondary.setNoOfScreen(screens);
System.out.println("Ente the ticket cost");
cost = sc.nextDouble();
secondary.setTicketCost(cost);
System.out.println("Enter the query id");
String queryid = sc.next();
query.setQueryId(queryid);
sc.nextLine();
System.out.println("Enter the query category");
String querycategory = sc.next();
query.setQueryCategory(querycategory);
sc.nextLine();
query.setPrimaryDataset(primary);
query.setSecondaryDataSet(secondary);
System.out.println(query);
```

```
        }


}
```

8) Retrieve Flights Based on Source and Destination

----------Flight.java* ----------

```
public class Flight {
private int flightId;
private String source;
private String destination;
private int noOfSeats;
private double flightFare;
public int getFlightId() {
return flightId;
}
public void setFlightId(int flightId) {
this.flightId = flightId;
}
public String getSource() {
return source;
}
public void setSource(String source) {
this.source = source;
}
```

```
public String getDestination() {
return destination;
}
public void setDestination(String destination) {
this.destination = destination;
}
public int getNoOfSeats() {
return noOfSeats;
}
public void setNoOfSeats(int noOfSeats) {
this.noOfSeats = noOfSeats;
}
public double getFlightFare() {
return flightFare;
}
public void setFlightFare(double flightFare) {
this.flightFare = flightFare;
}
public Flight(int flightId, String source, String destination,
int noOfSeats, double flightFare) {
super();
this.flightId = flightId;
this.source = source;
this.destination = destination;
this.noOfSeats = noOfSeats;
```

```
this.flightFare = flightFare;
}
}
```

----------FlightManagement.java* ----------

```
import java.util.ArrayList;
import java.sql.*;
public class FlightManagementSystem {
public ArrayList<Flight> viewFlightBySourceDestination(String source, String destination){
ArrayList<Flight> flightList = new ArrayList<Flight>();
try{
Connection con = DB.getConnection();
String query="SELECT * FROM flight WHERE source= '" + source + "' AND destination= '" +
destination + "' ";
Statement st=con.createStatement();
ResultSet rst= st.executeQuery(query);
while(rst.next()){
int flightId= rst.getInt(1);
String src=rst.getString(2);
String dst=rst.getString(3);
int noofseats=rst.getInt(4);
double flightfare=rst.getDouble(5);
flightList.add(new Flight(flightId, src, dst, noofseats, flightfare));
```

```
}
}catch(ClassNotFoundException | SQLException e){
e.printStackTrace();
}
return flightList;
}
}
```

----------DB.java* ----------

```
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
public class DB {
private static Connection con = null;
private static Properties props = new Properties();
//ENSURE YOU DON'T CHANGE THE BELOW CODE WHEN YOU SUBMIT
public static Connection getConnection() throws ClassNotFoundException, SQLException {
try{
FileInputStream fis = null;
fis = new FileInputStream("database.properties");
```

```
props.load(fis);

// load the Driver Class

Class.forName(props.getProperty("DB_DRIVER_CLASS"));

// create the connection now

con =
DriverManager.getConnection(props.getProperty("DB_URL"),props.getProperty("DB_USERN
AME"),props.getProperty("DB_PASSWORD"));

}

catch(IOException e){

e.printStackTrace();

}

return con;

}

}
```

----------Main.java* ----------

```
import java.util.Scanner;

import java.util.ArrayList;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter the source");

String source=sc.next();

System.out.println("Enter the destination");

String destination=sc.next();

FlightManagementSystem fms= new FlightManagementSystem();
```

```
ArrayList<Flight> flightList=fms.viewFlightBySourceDestination(source,destination);
if(flightList.isEmpty()){
System.out.println("No flights available for the given source and destination");
return;
}
System.out.println("Flightid Noofseats Flightfare");
for(Flight flight : flightList){
System.out.println(flight.getFlightId()+" "+flight.getNoOfSeats()+" "+flight.getFlightFare());
}
}
}
```