## 1. **Vehicle-Loan-Insurance - Use Interface**

Universal Loan and Insurance Providers is one of the fastest growing organization.   It needs to automate the transactions performed in the organization.  They need to automate the process of issuing loan and insurance coverage for vehicles.

**Partial code is provided to do this. Don’t change the skeleton. Do the additions wherever necessary.**

You are provided with a public class Vehicle with private attributes :

                      String vehicleNumber

                      String modelName

                      String vehicleType

                      double price

Appropriate public getters and setters are already written.

You are also provided with a 4 argument constructor with arguments –vehicleNumber, modelName, vehicleType and price.

**Note that the vehicleType can take the values as “4 wheeler” or “3 wheeler” or “2 wheeler”.**

**Write a public interface Loan  with an abstract method “double issueLoan()”.**

**Write a public interface Insurance with an abstract  method “double takeInsurance ()”.**

**The above class Vehicle should implement the Interfaces Loan and Insurance.**

Provide the implementation for issueLoan method based on the type of Vehicle.

            If the vehicleType is “4 wheeler”,  the eligible loan amount is 80% of its price.

            If the vehicleType is “3 wheeler”,  the eligible loan amount is 75% of its price.

            If the vehicleType is “2 wheeler”,  the eligible loan amount is 50% of its price.

Provide the implementation for takeInsurance based on price of vehicle.

            If the vehicle price is less than or equal to 150000 insurance amount is 3500.

            If the vehicle price is greater than 150000 and less than or equal to 300000 insurance amount is 4000.

            If the vehicle price is greater than 300000 insurance amount is 5000.

You are provided with a public class Main which has the main method.

Check the correctness of the methods written in these classes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Global Engineering is one of the fastest growing company. It needs to automate the transactions performed in the organization.  As start up, they need to automate the Employee management system.

**.**

You are provided with a public class Employee with protected attributes :

                                  int employeeId

                                  String employeeName

                                   double salary

       Appropriate public getters and setters are already written.

**Write a public 2 argument constructor with arguments – employeeId,and employeeName.**

**Write a public abstract method calculateSalary()  in Employee class as,**

**public abstract void calculateSalary()**

You are provided with a public class PermanentEmployee with private attribute :

                double basicPay

       Appropriate public getters and setters are already written.

***Make this class PermanentEmployee to inherit the Employee class.***

**Write a public 3 argument constructor with arguments – employeeId, employeeName and basicPay.**

**Implement the calculateSalary method in Employee class as**

**salary = basicPay – PF amount**

**Here PF Amount = basicPay \* 0.12; Set this value to the salary attribute.**

You are provided with a public class TemporaryEmployee with private attribute :

                int  hoursWorked

                int hourlyWages

     Appropriate public getters and setters are already written.

***This class TemporaryEmployee should inherit the Employee class.***

**Write a public 4 argument constructor with arguments – employeeId, employeeName, hoursWorked and hourlyWages.**

**Implement the calculateSalary method in Employee class as**

**salary = hoursWorked \* hourlyWages**

**Set this value to the salary attribute.**

You are provided with a public class Loan

A method calculateLoanAmount is provided as shown below :

public double calculateLoanAmount(Employee employeeObj)

**This method should calculate the loan amount and return that amount.**

**Provide the implementation for this method as mentioned below**

**Loan amount is calculated as follows :**

**If the Employee object is of type PermanentEmployee the loan amount should be 15%  of the salary.**

**If the Employee object is of type TemporaryEmployee the loan amount should be 10%  of the salary.**

You are provided with a public class Main which has the main method.

**Check the correctness of the methods written in these classes.**

**Note :  All class, methods needs to be declared as public**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Exception Handling:**

## **Array Manipulation - Use try with multi catch**

Tom wants to store the price details of the products that he purchased from the departmental store.  Help him do this by using the concept of Arrays.

To do this create a public class ArrayException with a method getPriceDetails as :

public String getPriceDetails() -  This method should do the following

Get the size of an array as input and then get the elements of the array(all elements are int) as input.

Next, user should provide the index of the array. This method should return the element at that index as “The array element is “+<that value>

This program may generate ArrayIndexOutOfBoundsException / InputMismatchException

In case of ArrayIndexOutOfBoundsException, the function should return “Array index is out of range”.

When providing the input, if the input is not an integer, it will generate InputMismatchException.  In this case the function should return “Input was not in the correct format”.

Use exception handling mechanism to handle the exception. Use separate catch block for handling each exception. In the catch block, return the appropriate message.

Write a main method and test the above function.

**Sample Input 1:**

Enter the number of elements in the array  
5  
Enter the price details  
50

80

60

70

40  
Enter the index of the array element you want to access  
1

**Sample Output 1:**

The array element is 80

**Sample Input 2:**

Enter the number of elements in the array  
2  
Enter the price details  
50  
80  
Enter the index of the array element you want to access  
9

**Sample Output 2:**

Array index is out of range

**Sample Input 3:**

Enter the number of elements in the array  
2  
Enter the price details  
30

j

**Sample Output 3:**

Input was not in the correct format

2. Andrew wants to teach division of two numbers to his son. To make his kid understand it better, he wants to write a program that divides two numbers and displays the answer.

Help him do this by writing a java program.

Write a public **class Division**.  Write a method divideTwoNumbers as

**public String divideTwoNumbers(int number1,int number2) – This method should perform division as number1 / number2.**

**If number2 is zero, it will throw ArithmeticException.**

**Whether division is done successfully or not, it should concatenate the String as “Thanks for using the application.”.**

When division done successfully, it should return a message as

“The answer is <number1/number2>. Thanks for using the application.”

If it results in ArithmeticException, it should return a message as

“Division by zero is not possible. Thanks for using the application.”

**Use try, catch and finally to perform the above task.**

Write the main method and test the above function.

**Sample Input 1:**

Enter the numbers

15

3

**Sample Output 1:**

The answer is 5. Thanks for using the application.

**Sample Input 2:**

Enter the numbers

15

0

**Sample Output 2:**

Division by zero is not possible. Thanks for using the application.

**Sample Input 3:**

Enter the numbers

15

2

**Sample Output 1:**

The answer is 7. Thanks for using the application.

3. Geneva Technologies is planning to conduct a Walk-in interview. The interview has 4 levels. To attend the interview, the candidates need to register the following information:  
Name, Gender and Expected salary.

Help him do this by writing a java program.

Partial code is provided.

You are provided with a public class Candidate with private attributes :

                        String name

                        String gender

                        double expectedSalary

            Appropriate getter and setters are provided.

            You are provided with a public class Main.

Write a method getCandidateDetails as –

            public static Candidate getCandidateDetails() – This method should get the candidate details, create the Candidate object using those details and return that object.

If the candidate’s expected salary is less than 10000

               ·   throw a user defined exception as InvalidSalaryException with the message “Registration Failed. Salary cannot be less than 10000.” and return null.

               ·    this method should throw / propagate InvalidSalaryException.

To do this, write a class InvalidSalaryException that inherits Exception class.

Write a constructor that takes a String as argument and set this string to the message attribute of the super class, Exception.

In the Main class, write the main method and test the method getCandidateDetails.

If it returns a valid Candidate object, then  display  “Registration Successful”.

Use a catch block to handle the exception that is returned by the method getCandidateDetails. In catch block display the message by using the getMessage() method.

**Sample Input 1:**

Enter the candidate Details

Name

Margrett

Gender

Female

Expected Salary

50000

**Sample Output 1:**

Registration Successful

**Sample Input 2:**

Enter the candidate Details

Name

Robin

Gender

Male

Expected Salary

5000

**Sample Output 2:**

Registration Failed. Salary cannot be less than 10000.