Q1:

(2 Point) Design and code a class named **`Teacher`** that holds information of a Teacher.

Teacher
-name:String -salary:double
+Teacher () +Teacher (name:String,salary:double) +getName():String +getSalary():double +setSalary(salary:double):void +toString():String

1. getName(): String – return a name of the teacher where every letters are in uppercase.

2. getSalary(): double – return a salary of the teacher

3. setSalary(salary:double): void – set current salary of the teacher is to a given salary.

4. toString():String – return value of a Teacher as a string *name salary*

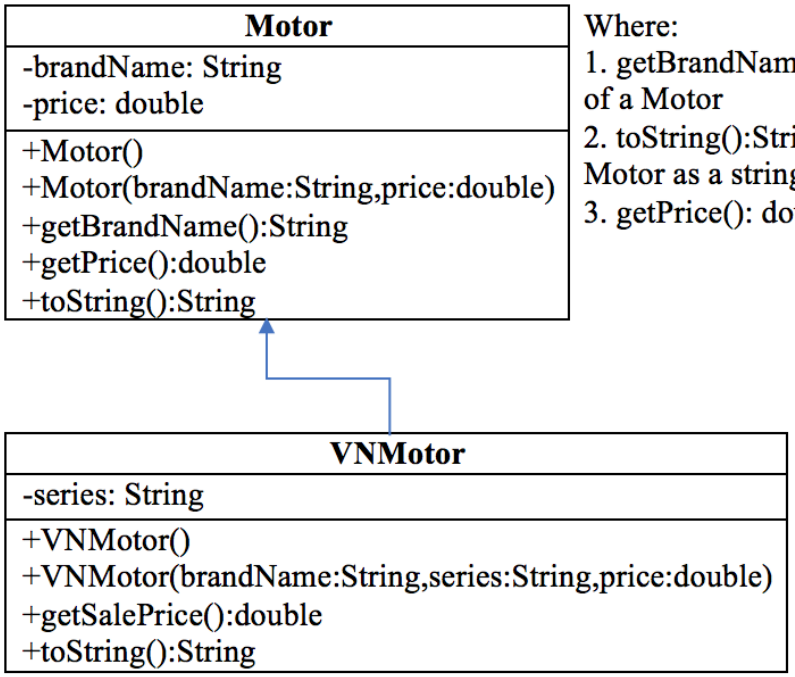
Do not format the result

The program output might look something like:

No of test case	**Correct output**
1	Enter Teacher name: Anton Enter Teacher salary: 2500 1. TC = 1 – test getName() 3. TC = 2 – test setSalary() 3. TC = 3 – test toString() Enter TC: 1 OUTPUT: ANTON
2	Enter Teacher name: Anton Enter Teacher salary: 2500 1. TC = 1 – test getName() 3. TC = 2 – test setSalary() 3. TC = 3 – test toString() Enter TC: 2 Enter new salary: 3700 OUTPUT: 3700.0
3	Enter Teacher name: Anton Enter Teacher salary: 2500 1. TC = 1 – test getName() 3. TC = 2 – test setSalary() 3. TC = 3 – test toString() Enter TC: 3 OUTPUT: Anton 2500.0

Q2:

(3 Point) Design and code a class named **`Motor`** that holds information about a **`Motor and`** class named **`VNMotor`** which is derived from **`Motor`**.



Where:
1. getBrandName():String – return brand name of a Motor
2. toString():String – return information of a Motor as a string as format of *brandName price*
3. getPrice(): double – return price of a Motor

- toString():String - return information of a VNMotor as a string as format of *brandName series price*
- getSalePrice():double – use to determine sale fare of a Motor, *sale price = original price – discount*, where:
    - discount = 5 percent out of original price if original price < 3000.
    - otherwise discount = 10 percent out of original price.

  Do not format the format the result.

The program output might look something like:

Correct output	Correct output

```
Enter brand name of a motor: Honda Future
Enter series of a motor: FX500
Enter price of a motor: 1300
1. TC = 1 – test toString function
2. TC = 2 – test getSalePrice function
3. TC = 3 – test getBrandName function
Enter TC: 2
OUTPUT:
1235.0
```

```
Enter brand name of a motor: Honda Future
Enter series of a motor: FX500
Enter price of a motor: 3000
1. TC = 1 – test toString function
2. TC = 2 – test getSalePrice function
3. TC = 3 – test getBrandName function
Enter TC: 2
OUTPUT:
2700.0
```

```
Enter brand name of a motor: Honda Future
Enter series of a motor: FX500
Enter price of a motor: 1300
1. TC = 1 – test toString function
2. TC = 2 – test getSalePrice function
3. TC = 3 – test getBrandName function
Enter TC: 1
OUTPUT:
Honda Future    1300.0
Honda Future    FX500   1300.0
```

```
Enter brand name of a motor: Honda Future
Enter series of a motor: FX500
Enter price of a motor: 1400
1. TC = 1 – test toString function
2. TC = 2 – test getSalePrice function
3. TC = 3 – test getBrandName function
Enter TC: 3
OUTPUT:
Honda Future
```

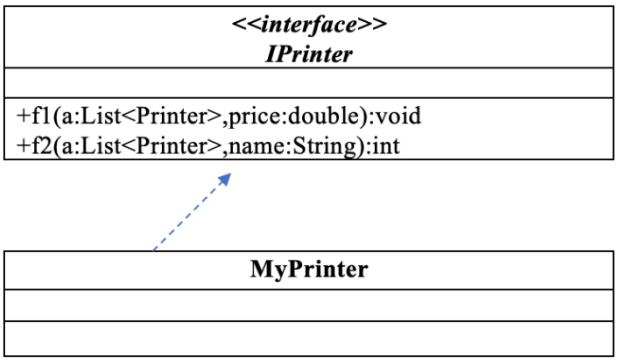
Q3:

(3 Point)
Design and code a class named Printer that holds information about a Printer.

Printer
-price:double -name:String
+Printer() +Printer(name:String,price:double) +getName():String +getPrice():double

Where:
1. getName():String – return name of a Printer
2. getPrice():double– return price of a Printer

The interface IPrinter is given (DO NOT CREATE THIS ONE). Design and code a class named **MyPrinter** which will implement interface IPrinter and complete 2 methods which were declared in IPrinter:



- void f1(List<Printer> a, double price) – remove from the list of printers "a" all printers which has price less than or equals to given price.
- int f2(List<Printer> a, String name) - count and return number of printers which are in the list "a" and has name contains given name. *The comparison must ignores the case during comparison.*

Given some data which is added to list "a" in the Main already:

Printer name	**Printer price**
HP 200J	110
HP 2000G	150
Canon G1240	120

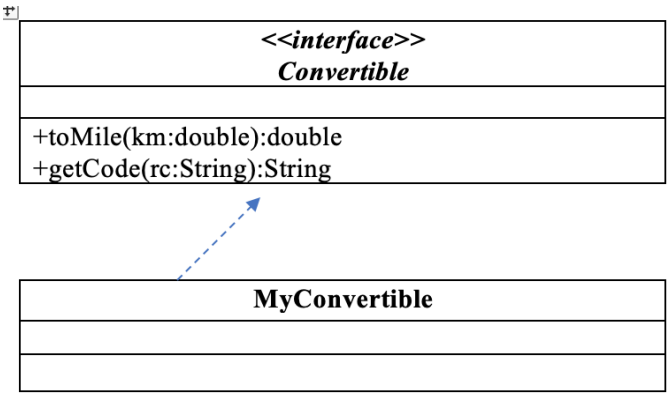
By using given data, the program output might look something like:

```
Add more how many printer: 1

Printer name: Canon PX2100
Printer price: 180.0

Enter test function (1-f1;2-f2): 1
Enter given printer price: 130

OUTPUT:
HP 2000G
Canon PX2100
```

```
Add more how many printer: 1

Printer name: Canon PX2100
Printer price: 180.0

Enter test function (1-f1;2-f2): 2
Enter given printer name: Canon

OUTPUT:
2
```

Q4:

(2 Point) You are given an interface named **Convertible (DO NOT CREATE THIS ONE)**. Design a class **MyConvertible** which will implement the interface Convertible.



1. double toMile(km:double) – used to convert a km value to mile value. *Given 1km = 0.621371 mile*

2. String getCode(rc:String) – assuming that length of Reservation code is dividable by 3 ; this function return code of RC as the rule:

- Code of RC = separate a RC into groups, each group has exactly 3 characters with same order of orginal RC, groups are seperated by character "-", eg A12-BE2-CM5

The program output might look something like:

No of test case	Correct output
1	1. TC = 1 – test toMile() 2. TC = 2 – test getCode() Enter TC: 1 Enter a value in km: 16.5 OUTPUT: 10.25
2	1. TC = 1 – test toMile() 2. TC = 2 – test getCode() Enter TC: 2 Enter a value of rc: K2M762 OUTPUT: K2M-762