Question 1:

(3 marks) You have main function which is saved on folder named Q1 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`Employee`** that holds information about a **`Employee`**. Information of an **`Employee`** includes:

- A string describes the name of an Employee.
- A double value describes basic salary of an Employee.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class.

Design and code a deriving class named **`Seller`** from **`Employee`** class that holds information about a **`Seller`**. Information of a **`Seller`** includes:

- A double value describes revenue of a **Seller** of a year.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run and complete the below method which is declared in Seller class, the function will be used in second test case.
    - double getSalary() – use to determine sale price of a seller; *salary = basic salary + bonus*, where:
        - bonus = 5 percent out of revenue if revenue < 30000.
        - otherwise bonus = 10 percent out of revenue.

The program output might look something like:

No of test case	Correct output	Correct output
2	Enter employee name: Anton Enter employee salary: 1500 Enter employee revenue: 31000 Enter TC: 2 OUTPUT: 4600.0	Enter employee name: Anton Enter employee salary: 1500 Enter employee revenue: 29000 Enter TC: 2 OUTPUT: 2950.0

1	Enter employee name: Anton Enter employee salary: 1500 Enter employee revenue: 31000 Enter TC: 1 OUTPUT: Anton 1500.0 Anton 1500.0 31000.0	

Question 2:

(4 Point) You have main function which is saved on folder named Q2 already. Use these classes and function to test all your operations in this question, DO NOT EDIT ANY CODE in main. You can add more operations/classes/interfaces to be able to answer the below questions.
We had provided you:
1. Interface – ICourse which will declare some operations for a Course – DO NOT EDIT this one.
2. **You only need to complete the code in class MyCourse and Course.**
Design and code a class named Course that holds information about a Course. Information of a Course includes:

- A double value holding fee of a Course.
- A string describing name of a Course.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run and complete the function
    - String getName() – return name of a Course.

Design and code a class named **MyCourse** which will implement interface ICourse and complete 2 methods which were declared in ICourse

- void f1(List<Course> a, int st) – Sort the list of courses "a" ascending by course fee if st = 1 otherwise sort the list of courses "a" descending by course name. *The comparison must ignores the case during comparison.*
- int f2(List<Course> a, double fee) - count and return numbers of courses in the list "a" which are in the list "a" and has course fee greater than or equals given fee.

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

Course name	Course fee
PRJ311	110
DBI202	150
PRF192	120

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

```
Add more how many course: 1

Course name: PRN292
Course fee: 180.0

Enter test function (1-f1;2-f2): 1
Enter st: 1

OUTPUT:
PRJ311
PRF192
DBI202
PRN292
```

```
Add more how many course: 1

Course name: PRN292
Course fee: 180.0

Enter test function (1-f1;2-f2): 1
Enter st: 2

OUTPUT:
PRN292
PRJ311
PRF192
DBI202
```

```
Add more how many course: 1

Course name: PRN292
Course fee: 180.0

Enter test function (1-f1;2-f2): 2
Enter course fee: 150

OUTPUT:
2
```

Question 3:

(3 Point) You have main function which is saved on folder named Q3 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`ReservationCode`** that holds information of a ReservationCode (RC) for a customer, including:

- A string describes value of a customer name.
- A string describes value for a RC.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class and complete below method which is declared in ReservationCode class, the function will be used in the second test case.
  - String getCode() – 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-CMK

The program output might look something like:

No of test case	Correct output
2	Enter customer name: Anton Enter reservation code: K2M76A Enter TC: 2 OUTPUT: K2M-76A
1	Enter customer name: Anton Enter reservation code: K2M1D9 Enter TC: 1 OUTPUT: Anton K2M1D9