

Instructions

- You need to solve **one programming problem** in this challenge.
- The duration of this challenge is **120 Minutes**.
- Programming questions have a **Compile and Run** option where you can run your solution against sample test cases before submitting it.
- Click **Evaluate** button only if your code compiles successfully.
- This challenge covers the following topic(s).
 - Class and Objects
 - Functions
 - ADO.Net

Scenario:

The transport department of your company wants to have an application through which they can add employees commute records to database, display all commute records from database and also display records of all those employees commute that are paying the lowest commute fare. Help the transport department by creating the application.

Functionalities:

- Add employee commute records to database
- Display all employee commute records from database
- Display the records of those employees who are paying the lowest commute fare.

Data Design:

Tablename: tblOfficeCommuteApp		
Column Name	Data type	Constraints
Employee_Id	varchar(100)	Primary Key, Not Null
Employee_Name	varchar(100)	Not Null
Employee_Type	varchar(100)	Not Null
Travel_Distance	float	Not Null
Commute_Charge	float	Not Null

The database connection information is specified in the “App.config” file, which is also provided as part of code skeleton. **THIS IS GIVEN ONLY FOR YOUR REFERENCE.** You need **NOT** change this. The code skeleton will be available in the Tekstac platform.

Component Specification

- Create a model class called **OfficeCommuteDAO** with the below public auto-implemented properties:

Type (Class)	Data Types	Properties
OfficeCommuteDAO	string	EmployeeId
	string	EmployeeName
	string	EmployeeType
	float	TravelDistance
	float	CommuteCharge

Also create a default constructor and a parameterized constructor under this class.

- Create a Class called **OfficeCommuteBO** with the below methods:

Type(Class)	Methods	Responsibilities
OfficeCommuteBO	public int InsertRecords(OfficeCommuteDAO commuteData)	This method should accept an OfficeCommuteDAO object and execute a sql query to insert an employee commute record into the database. It returns 1 if the insertion is successful and 0 if it fails to insert.
	public IList< OfficeCommuteDAO > DisplayAllRecords()	This method will retrieve all commute records from the database. Store each record in OfficeCommuteDAO object. Add these objects to a 'List'.
	public IList< OfficeCommuteDAO > DisplayRecordsByLowestCommuteFare()	This method will retrieve all those employees commute records from the database whose is paying the lowest fare. Store each record in OfficeCommuteDAO object. Add these objects to a 'List'.

- Create a class called **Program** with main method. Call the **OfficeCommuteBO** methods under the Program class and test your application.

Fare Calculation Formula:

Employee Type	Travel Distance	Fare Calculation
INTERNAL	<=10	commuteFare = travelDistance * 70
	>10 and <=20	commuteFare = travelDistance * 90
	>20	commuteFare = travelDistance * 110
EXTERNAL	<=10	commuteFare = travelDistance * 110
	>10 and <=20	commuteFare = travelDistance * 130
	>20	travelDistance = travelDistance * 150

Business Rules:

1. Employee Id should be in the format: The first 3 should be letters (either EXT or INT only) and the next 4 should be numbers (example: EXT1001 is valid, INT001 is valid where as VEN1001 is invalid, ext1001 is invalid, int1001 is invalid). If validation fails return false; else return true
2. Duplicate Employee Id should not be present in the data table. If same Id already exists in table return false; else return true
3. Employee Type should be either INTERNAL or EXTERNAL. If any other value is entered return false; else return true.
4. The Employee Type first 3 letters should be checked if it matches with the first 3 letters of Employee Id. If data matches return true; else return false. [Note: The match checking should be case insensitive].
5. The travel distance should be between 2 km and 30 km. If value is entered outside the specified range return false; else return true.
6. The fare should be rounded to 2 decimal places for display.
7. Enclose all DML and DQL operations in **OfficeCommuteBO** class under try-catch block.

Sample Input / Output

```
Welcome Admin to Office Commute System

Menu:
Enter 1 to Add Employee Commute Records
Enter 2 to Display All Employee Commute Records
Enter 3 to Display All Employee Commute Records Who Are Paying Lowest Fare
1
Enter Employee Id:
INT1001
Enter Employee Name:
Sam
Enter Employee Type(External or Internal):
INTERNAL
Enter Travel Distance:
5
Data Insertion Successful

Press yes to continue your work..Any other key to terminate operation
yes

Menu:
Enter 1 to Add Employee Commute Records
Enter 2 to Display All Employee Commute Records
Enter 3 to Display All Employee Commute Records Who Are Paying Lowest Fare
1
Enter Employee Id:
EXT1001
Enter Employee Name:
Joe
Enter Employee Type(External or Internal):
```

```

EXTERNAL
Enter Travel Distance:
8
Data Insertion Successful

Press yes to continue your work..Any other key to terminate operation
yes

Menu:
Enter 1 to Add Employee Commute Records
Enter 2 to Display All Employee Commute Records
Enter 3 to Display All Employee Commute Records Who Are Paying Lowest Fare
1
Enter Employee Id:
EXT1002
Enter Employee Name:
Merry
Enter Employee Type(External or Internal):
EXTERNAL
Enter Travel Distance:
3
Data Insertion Successful

Press yes to continue your work..Any other key to terminate operation
yes

Menu:
Enter 1 to Add Employee Commute Records
Enter 2 to Display All Employee Commute Records
Enter 3 to Display All Employee Commute Records Who Are Paying Lowest Fare

```

```

2

Display All Employees Commute Infomation:
Employee Id      Name      Employee Type  Distance      Commute Fare
EXT1001          Joe      EXTERNAL       8              880.00
EXT1002          Merry    EXTERNAL       3              330.00
INT1001          Sam      INTERNAL       5              350.00

Press yes to continue your work..Any other key to terminate operation
yes

Menu:
Enter 1 to Add Employee Commute Records
Enter 2 to Display All Employee Commute Records
Enter 3 to Display All Employee Commute Records Who Are Paying Lowest Fare
3

Display Employees With Lowest Commute Fare:
Employee Id      Name      Employee Type  Distance      Commute Fare
EXT1002          Merry    EXTERNAL       3              330.00

Press yes to continue your work..Any other key to terminate operation
no

Thank you for using the app. Have a nice day

```