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 highest 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 highest 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 **OfficeCommuteDataInfo** with the below public auto-implemented properties:

Type (Class)	Data Types	Properties
OfficeCommuteDataInfo	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 **OfficeCommuteDataOperations** with the below methods:

Type(Class)	Methods	Responsibilities
OfficeCommuteDataOperations	public bool AddRecords(OfficeCommuteDataInfo commuteData)	This method should accept a **OfficeCommuteDataInfo** object and execute a sql query to insert a employee commute record into the database. It returns true if the insertion is successful and false if it fails to insert.
	public IList<OfficeCommuteDataInfo> DisplayRecords()	This method will retrieve all commute records from the database. Store each record in OfficeCommuteDataInfo object. Add these objects to a 'List'.
	public IList<OfficeCommuteDataInfo> DisplayRecordsByHigestCommuteFare()	This method will retrieve all those employees commute records from the database whose is paying the highest fare. Store each record in OfficeCommuteDataInfo object. Add these objects to a 'List'. ***[Hint: More than one employee can have the highest fare; so use a sub query for this method].***

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

Fare Calculation Formula:

Employee Type	Travel Distance	Fare Calculation
INTERNAL	<=10	commuteFare = travelDistance * 50
	>10 and <=20	commuteFare = travelDistance * 60
	>20	commuteFare = travelDistance * 70
EXTERNAL	<=10	commuteFare = travelDistance * 70
	>10 and <=20	commuteFare = travelDistance * 80
	>20	travelDistance = travelDistance * 90

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.

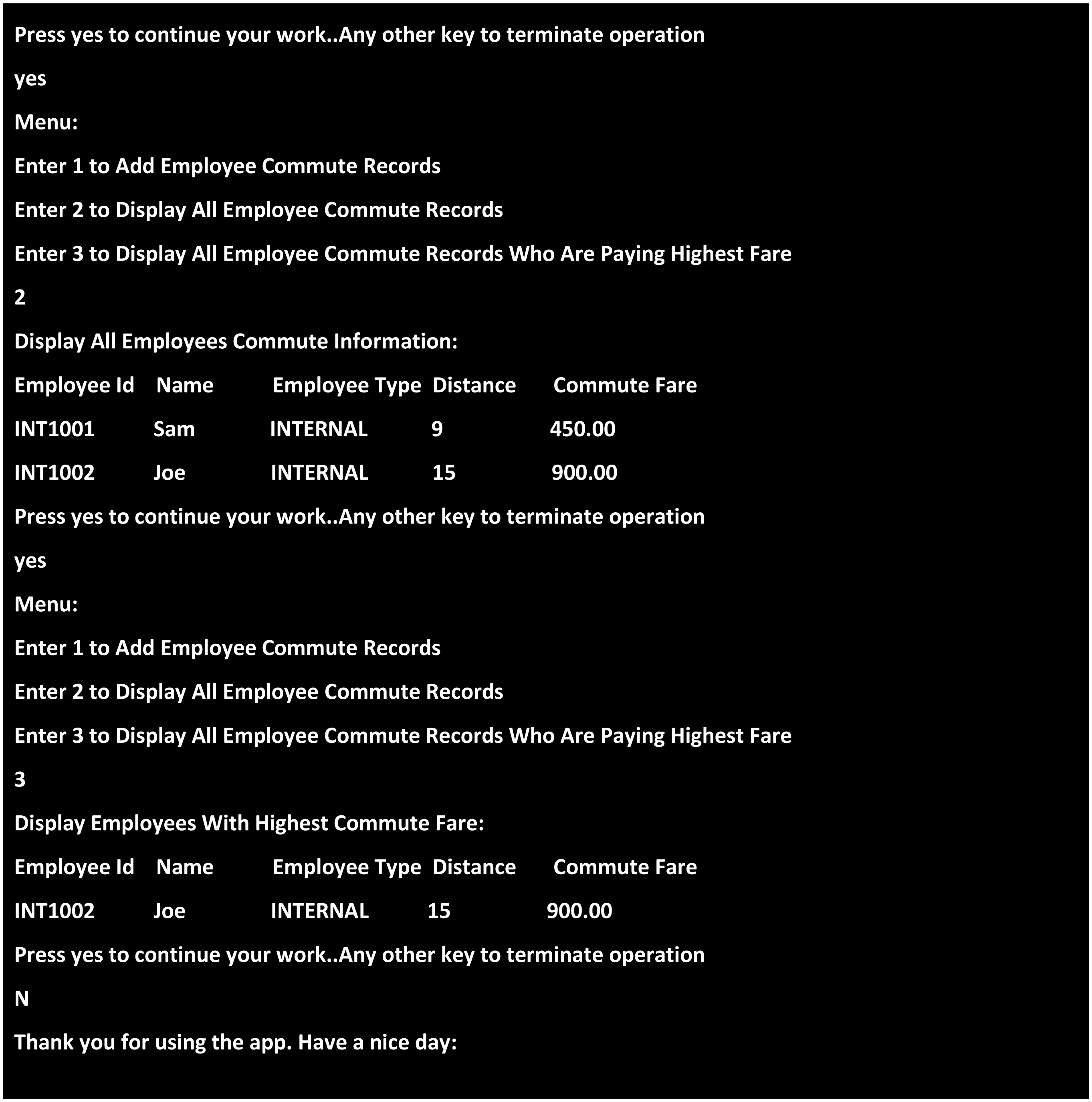
6. The fare should be rounded to 2 decimal places for display.

7. Enclose all DML and DQL operations in OfficeCommuteDataOperations 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 Highest Fare
1
Enter Employee Id:
int1001
Wrong Employee Id Entered..Try Again
Enter Employee Id:
INT1001
Enter Employee Name:
Sam
Enter Employee Type(External or Internal):
vendor
Invalid Employee Type..Try Again
Enter Employee Type(External or Internal):
```

```
EXTERNAL
Employee Type does not match with Employee Id type..Try Again
Enter Employee Type(External or Internal):
INTERNAL
Enter Travel Distance:
100
Invalid Travel Distance..Try Again
Enter Travel Distance:
1
Invalid Travel Distance..Try Again
Enter Travel Distance:
9
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 Highest Fare
1
Enter Employee Id:
INT1001
Same Employee Id already present in database..Try With another ID
Enter Employee Id:
INT1002
Enter Employee Name:
Joe
Enter Employee Type(External or Internal):
INTERNAL
Enter Travel Distance:
15
```



```
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 Highest Fare
2
Display All Employees Commute Information:
Employee Id    Name          Employee Type  Distance      Commute Fare
INT1001        Sam           INTERNAL       9             450.00
INT1002        Joe           INTERNAL       15            900.00
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 Highest Fare
3
Display Employees With Highest Commute Fare:
Employee Id    Name          Employee Type  Distance      Commute Fare
INT1002        Joe           INTERNAL       15            900.00
Press yes to continue your work..Any other key to terminate operation
N
Thank you for using the app. Have a nice day:
```