Company Registration

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Registration   //DO NOT change the namespace name
{
   public class Program   //DO NOT change the class name
    {
       public static void Main(string[] args)   //DO NOT change the 'Main' method
signature
        {

            //Implement the code here
            Employee E = new Employee();
            Company C =new Company();
            CompanyContext context = new CompanyContext();
            CompanyUtil cutil = new CompanyUtil();

            Console.WriteLine("Enter Employee Id");
            E.EmployeeId = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Employee Name ");
            E.EmployeeName = Console.ReadLine();
            Console.WriteLine("Enter Experience");
            E.Experience = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Salary");
            E.Salary = Convert.ToDouble(Console.ReadLine());
            cutil.AddEmployee(E);
            Console.WriteLine("Employee Inserted Successfully ");

             Console.WriteLine("Enter Employee Id");
            E.EmployeeId = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Employee Name ");
            E.EmployeeName = Console.ReadLine();
            Console.WriteLine("Enter Experience");
            E.Experience = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Salary");
            E.Salary = Convert.ToDouble(Console.ReadLine());
            cutil.AddEmployee(E);
            Console.WriteLine("Employee Inserted Successfully ");

            Console.WriteLine("Enter Company Name ");
            C.CompanyName = Console.ReadLine();
            Console.WriteLine("Enter Employee Name ");
            E.EmployeeName = Console.ReadLine();
            Console.WriteLine("Enter Date of Join ");
            C.DateOfJoin = Convert.ToDateTime(Console.ReadLine());
            Console.WriteLine("Enter City ");
            C.City = Console.ReadLine();
            cutil.AddCompany(C);
            Console.WriteLine("Company Inserted Successfully ");

            Console.WriteLine("Enter Company Name ");
            C.CompanyName = Console.ReadLine();
            Console.WriteLine("Enter Employee Name ");
            E.EmployeeName = Console.ReadLine();
            Console.WriteLine("Enter Date of Join ");
```

```
            C.DateOfJoin = Convert.ToDateTime(Console.ReadLine());
            Console.WriteLine("Enter City ");
            C.City = Console.ReadLine();
            cutil.AddCompany(C);
            Console.WriteLine("Company Inserted Successfully ");

            Console.WriteLine("Retrieve all  Company  based on Employee Name");
            Console.WriteLine("Enter Employee Name");
            string en=Console.ReadLine();
            var res=cutil.GetCompanyByEmployeeName(en);
            foreach(Company ecn in res){
                Console.WriteLine("{0}",ecn.CompanyName.ToString());
            }
        }
    }
}


----------------------------------Company.cs--------------------------------------
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;

namespace Registration
{
    class Company
    {
        public int CompanyId { get; set; }
        public String CompanyName { get; set; }
        public DateTime DateOfJoin { get; set; }
        public String City { get; set; }

         // Add 2 properties
        //1. Include a reference navigation property of Employee type
        //2. foreign key property of EmployeeName
        public virtual Employee Employee{get;set;}
        public virtual string EmployeeName{get; set;}
    }
}

----------------------------------Employee.cs--------------------------------------
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Registration
{
    class Employee
    {
        public int EmployeeId { get; set; }
        public String EmployeeName { get; set; }
        public int Experience { get; set; }
        public double Salary { get; set; }
        //Include a collection navigation property of type ICollection<Employee>
         public virtual ICollection<Company>Company{get;set;}
    }
}
```

```
---------------------------------CompanyContext.cs-----------------------------------
--
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.Entity;

namespace Registration  //DO NOT change the namespace name
{
    class CompanyContext: DbContext   //DO NOT change the class name
    {
         //Implement property for 'Companys' and 'Employees' with required declaration
        public virtual DbSet<Company>Companys{get;set;}
        public virtual DbSet<Employee>Employees{get;set;}

        public CompanyContext() : base("DataConnection")  //DO NOT change the Context
name
        {


        }
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
           //Map Student entity to EmployeeDetail table
            //Map Course entity to CompanyDetail table

            //Make 'EmployeeName' as Foreign key in Company Entity
            //configure one-to-many relationship as mentioned in the problem statement
            modelBuilder.Entity<Company>().ToTable("CompanyDetail");
            modelBuilder.Entity<Employee>().ToTable("EmployeeDetail");
            modelBuilder.Entity<Employee>().HasKey(E=> E.EmployeeName).Property(E =>
E.EmployeeName).IsRequired();
            modelBuilder.Entity<Company>().HasRequired<Employee>(E =>
E.Employee).WithMany(C => C.Company).HasForeignKey(E =>
E.EmployeeName).WillCascadeOnDelete();
        }
}
}

---------------------------------------CompanyUtil.cs---------------------------------
----------
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Registration  //DO NOT change the namespace name
{
    class CompanyUtil  //DO NOT change the class name
    {


        public Company AddCompany(Company company)  //DO NOT change method signature
        {
           //Implement the code here
           using(CompanyContext context=new CompanyContext()){
                context.Companys.Add(company);
                context.SaveChanges();
```

```
                return company;
            }
        }

        public Employee AddEmployee(Employee employee)  //DO NOT change method
signature
        {
           //Implement the code here
           using (CompanyContext context=new CompanyContext()){
                context.Employees.Add(employee);
                context.SaveChanges();
                return employee;
            }
        }

        public List<Company> GetCompanyByEmployeeName(string employeeName)  //DO NOT
change method signature
        {
            //Implement the code here
            using(CompanyContext context=new CompanyContext()){
                var companyList =context.Companys.Where(E =>
E.EmployeeName.Equals(employeeName,StringComparison.InvariantCultureIgnoreCase)).ToList
();
                return companyList;
            }
        }


    }
}
```