Indexers

An *indexer* is a member that enables objects to be indexed in the same way as an array. An indexer is declared like a property except that the name of the member is **this** followed by a parameter list written between the delimiters **[** and **].** The parameters are available in the accessor(s) of the indexer. Similar to properties, indexers can be read-write, read-only, and write-only, and the accessor(s) of an indexer can be virtual.

Indexers enable objects to be indexed in a similar manner to arrays. A `get` accessor returns a value. A `set` accessor assigns a value. The `this` keyword is used to define the indexer. The `value` keyword is used to define the value being assigned by the set accessor. Indexers do not have to be indexed by an integer value; it is up to you how to define the specific look-up mechanism. Indexers can be overloaded. Indexers can have more than one formal parameter, for example, when accessing a two-dimensional array.

Example:

```
using System.Collections.Generic;
using System.Linq;
namespace IndexersDemo
{
    public class Employee
    {
        public int EmployeeId { get; set; }
        public string Name { get; set; }
        public string Gender { get; set; }
        public double Salary { get; set; }
    }
    public class Company
    {
        private List<Employee> listEmployees;
        public Company()
        {
            listEmployees = new List<Employee>();

            listEmployees.Add(new Employee
            { EmployeeId = 101, Name = "Raj", Gender = "Male", Salary = 1000});

            listEmployees.Add(new Employee
            { EmployeeId = 102, Name = "Aarti", Gender = "Female", Salary = 2000});

            listEmployees.Add(new Employee
            { EmployeeId = 103, Name = "Jay", Gender = "Male", Salary = 5000});
        }
//The integer indexer take an employeeId as parameter and return the employee name
        public string this[int employeeId]
        {
            get
            {
                return listEmployees.
                    FirstOrDefault(x => x.EmployeeId == employeeId).Name;
            }
            set
            {
                listEmployees.
                    FirstOrDefault(x => x.EmployeeId == employeeId).Name = value;
            }
        }
//The string indexer that returns the total count of employees whose gender
//matches with the gender that is passed in and Changes the gender of all
//employees whose gender matches with the gender that is passed in.
        public string this[string gender]
        {
            get
            {
                return listEmployees.Count(x => x.Gender.ToLower() ==
gender.ToLower()).ToString();
            }
            set
            {

                foreach (Employee employee in listEmployees)
                {
                    if (employee.Gender == gender)
                    {
                        employee.Gender = value;
                    }
                }
            }
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Company company = new Company();

            Console.WriteLine("Name of Employee with Id = 101: " + company[101]);
            Console.WriteLine("Changing the name of employee with Id = 101");
            company[101] = "Raju";
            Console.WriteLine("Name of Employee with Id = 101: " + company[101]);

            Console.WriteLine("Before changing the Gender of all the male
employees to Female");

            Console.WriteLine("Total Number Employees with Gender = Male:" +
company["Male"]);
            Console.WriteLine("Total Number Employees with Gender = Female:" +
company["Female"]);
```

```
            company["Male"] = "Female";
            Console.WriteLine("After changing the Gender of all male employees to 
Female");
            Console.WriteLine("Total Employees with Gender = Male:" + 
company["Male"]);
            Console.WriteLine("Total Employees with Gender = Female:" + 
company["Female"]);
        }
    }
}
//Output:
//    Name of Employee with Id = 101: Raj
//    Changing the name of employee with Id = 101
//    Name of Employee with Id = 101: Raju
//    Before changing the Gender of all the male employees to Female
//    Total Number Employees with Gender = Male:2
//    Total Number Employees with Gender = Female:1
//    After changing the Gender of all male employees to Female
//    Total Employees with Gender = Male:0
//    Total Employees with Gender = Female:3
```