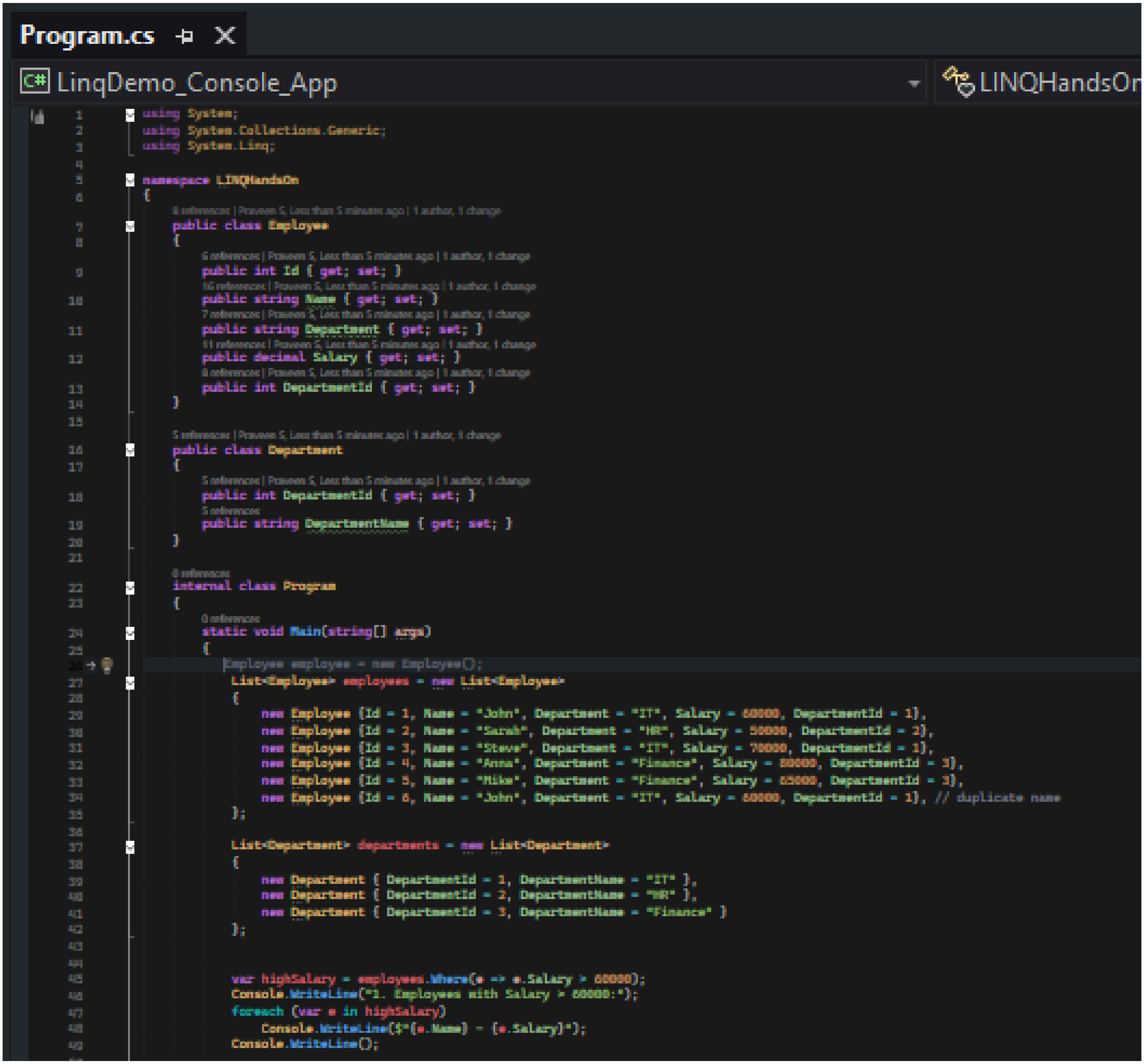
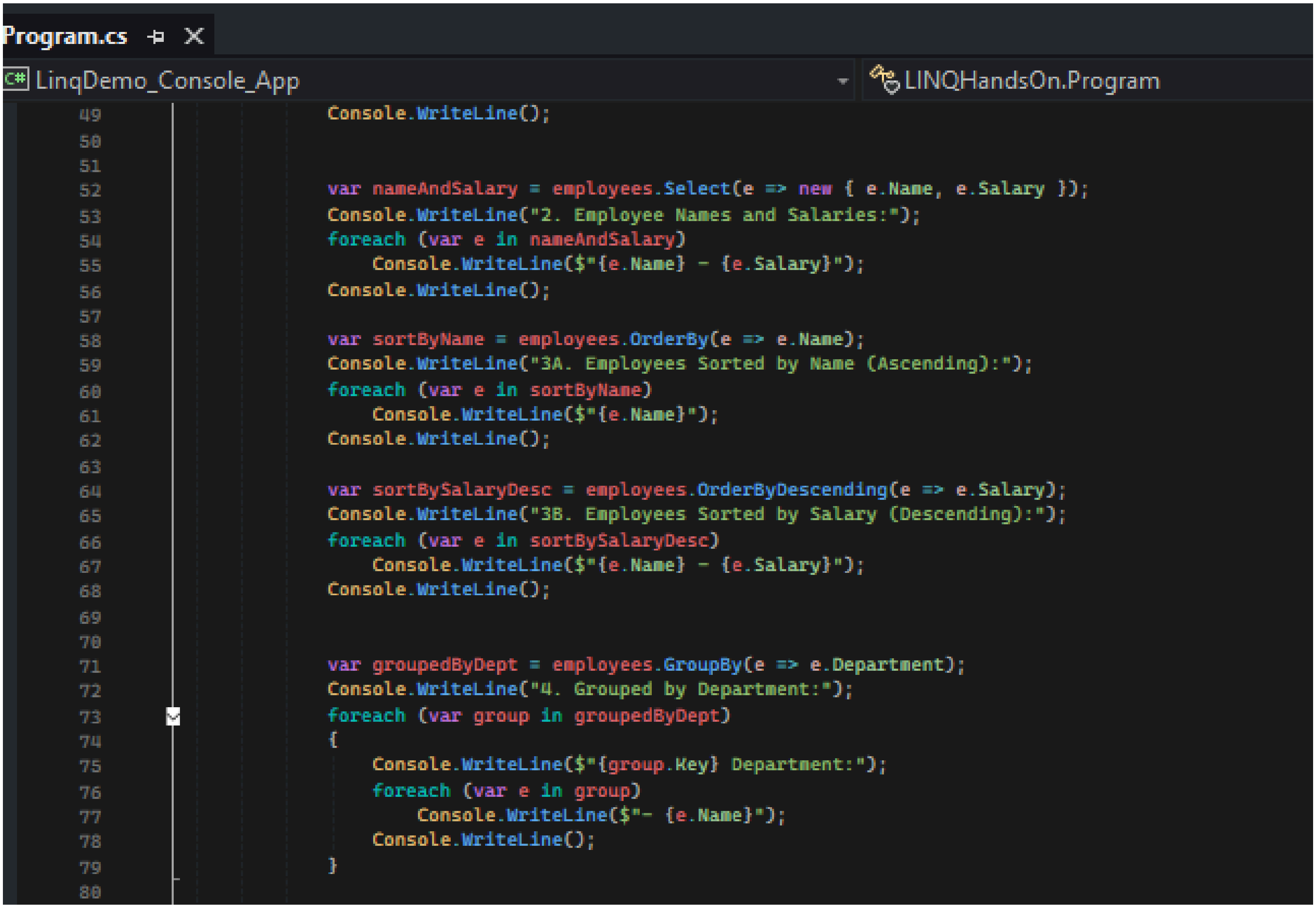
Assignment 7

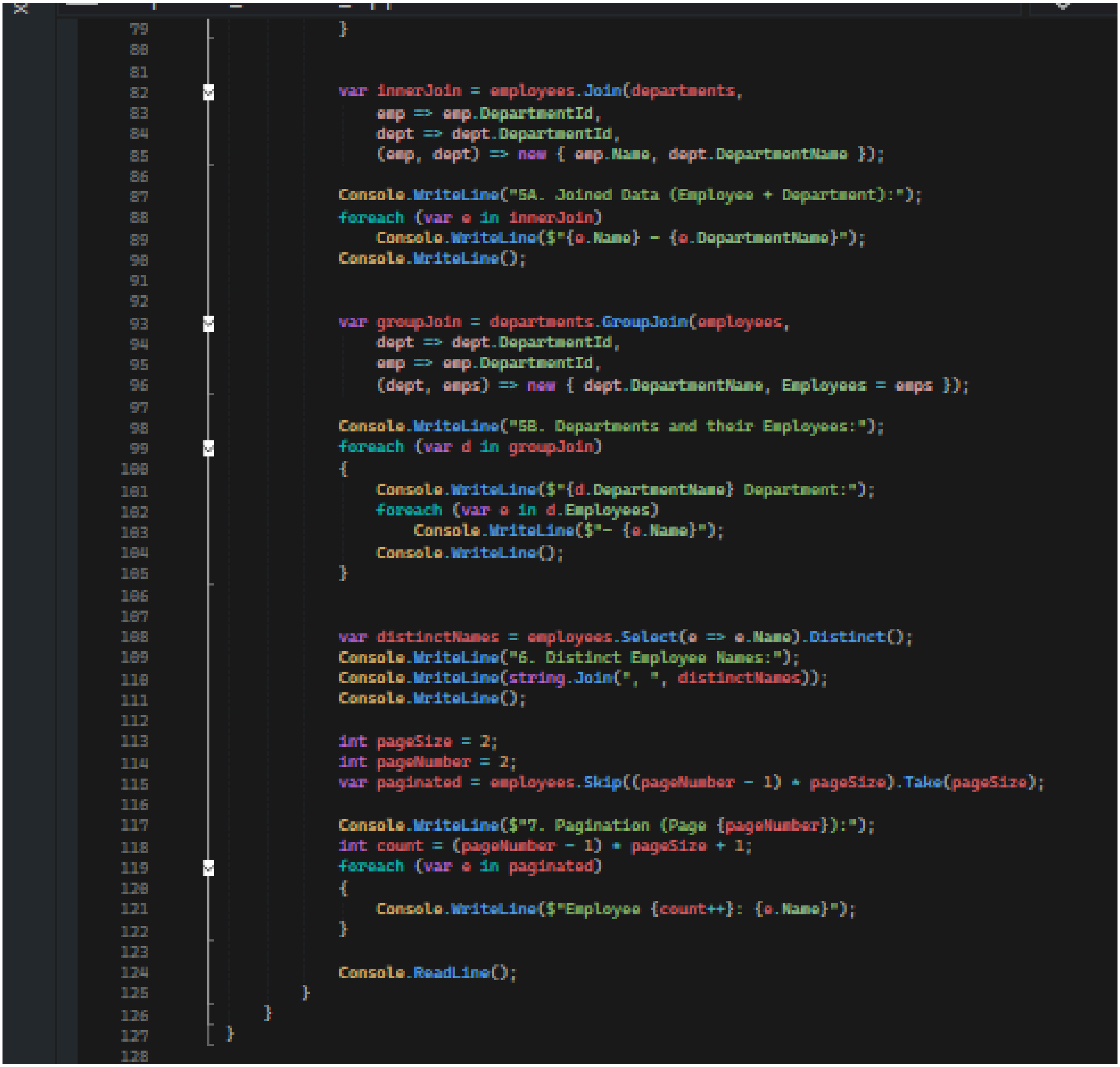
Linq Demo:



```csharp
using System;
using System.Collections.Generic;
using System.Linq;

namespace LINQHandsOn
{
    public class Employee
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Department { get; set; }
        public decimal Salary { get; set; }
        public int DepartmentId { get; set; }
    }

    public class Department
    {
        public int DepartmentId { get; set; }
        public string DepartmentName { get; set; }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Employee employee = new Employee();
             List<Employee> employees = new List<Employee>
             {
                 new Employee {Id = 1, Name = "John", Department = "IT", Salary = 60000, DepartmentId = 1},
                 new Employee {Id = 2, Name = "Sarah", Department = "HR", Salary = 50000, DepartmentId = 2},
                 new Employee {Id = 3, Name = "Steve", Department = "IT", Salary = 70000, DepartmentId = 1},
                 new Employee {Id = 4, Name = "Anna", Department = "Finance", Salary = 80000, DepartmentId = 3},
                 new Employee {Id = 5, Name = "Mike", Department = "Finance", Salary = 65000, DepartmentId = 3},
                 new Employee {Id = 6, Name = "John", Department = "IT", Salary = 60000, DepartmentId = 1}, // duplicate name
             };

             List<Department> departments = new List<Department>
             {
                 new Department { DepartmentId = 1, DepartmentName = "IT" },
                 new Department { DepartmentId = 2, DepartmentName = "HR" },
                 new Department { DepartmentId = 3, DepartmentName = "Finance" }
             };


             var highSalary = employees.Where(e => e.Salary > 60000);
             Console.WriteLine("1. Employees with Salary > 60000:");
             foreach (var e in highSalary)
                 Console.WriteLine($"{e.Name} - {e.Salary}");
             Console.WriteLine();
```



```
            Console.WriteLine();


            var nameAndSalary = employees.Select(e => new { e.Name, e.Salary });
            Console.WriteLine("2. Employee Names and Salaries:");
            foreach (var e in nameAndSalary)
                Console.WriteLine($"{e.Name} - {e.Salary}");
            Console.WriteLine();

            var sortByName = employees.OrderBy(e => e.Name);
            Console.WriteLine("3A. Employees Sorted by Name (Ascending):");
            foreach (var e in sortByName)
                Console.WriteLine($"{e.Name}");
            Console.WriteLine();

            var sortBySalaryDesc = employees.OrderByDescending(e => e.Salary);
            Console.WriteLine("3B. Employees Sorted by Salary (Descending):");
            foreach (var e in sortBySalaryDesc)
                Console.WriteLine($"{e.Name} - {e.Salary}");
            Console.WriteLine();


            var groupedByDept = employees.GroupBy(e => e.Department);
            Console.WriteLine("4. Grouped by Department:");
            foreach (var group in groupedByDept)
            {
                Console.WriteLine($"{group.Key} Department:");
                foreach (var e in group)
                    Console.WriteLine($"- {e.Name}");
                Console.WriteLine();
            }

```



```
            }


            var innerJoin = employees.Join(departments,
                emp => emp.DepartmentId,
                dept => dept.DepartmentId,
                (emp, dept) => new { emp.Name, dept.DepartmentName });

            Console.WriteLine("5A. Joined Data (Employee + Department):");
            foreach (var e in innerJoin)
                Console.WriteLine($"{e.Name} - {e.DepartmentName}");
            Console.WriteLine();


            var groupJoin = departments.GroupJoin(employees,
                dept => dept.DepartmentId,
                emp => emp.DepartmentId,
                (dept, emps) => new { dept.DepartmentName, Employees = emps });

            Console.WriteLine("5B. Departments and their Employees:");
            foreach (var d in groupJoin)
            {
                Console.WriteLine($"{d.DepartmentName} Department:");
                foreach (var e in d.Employees)
                    Console.WriteLine($"- {e.Name}");
                Console.WriteLine();
            }


            var distinctNames = employees.Select(e => e.Name).Distinct();
            Console.WriteLine("6. Distinct Employee Names:");
            Console.WriteLine(string.Join(", ", distinctNames));
            Console.WriteLine();

            int pageSize = 2;
            int pageNumber = 2;
            var paginated = employees.Skip((pageNumber - 1) * pageSize).Take(pageSize);

            Console.WriteLine($"7. Pagination (Page {pageNumber}):");
            int count = (pageNumber - 1) * pageSize + 1;
            foreach (var e in paginated)
            {
                Console.WriteLine($"Employee {count++}: {e.Name}");
            }

            Console.ReadLine();
        }
    }
}

```

Github Link:
https://github.com/praveen-dotnet-chn/LinqDemo_Console_App