Day 2:

05-08-2025

VIJAY M

Generic with Repo:

Program.cs

```
using System;

using Employee.Repo;

namespace Employee

{

    class Program

    {

        static void Main()

        {

            IEmpRepo<Employee> employeeRepo = new EmpRepo<Employee>();

            Employee emp1 = new Employee(1, "Alice", "alice@example.com", new DateOnly(2022, 1, 1), new DateOnly(1995, 5, 20));

            Employee emp2 = new Employee(2, "Bob", "bob@example.com", new DateOnly(2023, 3, 15), new DateOnly(1990, 8, 10));

            employeeRepo.Add(emp1);

            employeeRepo.Add(emp2);

            var allEmployees = employeeRepo.GetAll();

            foreach (var emp in allEmployees)

            {

                Console.WriteLine(emp);

            }

            Manager man1 = new Manager(3, "john", "john@example.com", new DateOnly(2020, 1, 1), new DateOnly(1996, 6, 2),10000);

            Manager man2 = new Manager(4, "newton", "newton@example.com", new DateOnly(2022, 3, 15), new DateOnly(1993, 8, 10),20000);

            IEmpRepo<Manager> managerRepo = new EmpRepo<Manager>();
```

```
            managerRepo.Add(man1);
            managerRepo.Add(man2);
            var allManager = managerRepo.GetAll();
            foreach (var emp in allManager)
            {
                Console.WriteLine(emp);
            }
        }
    }
}
```

Employee.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Employee
{
    internal class Employee
    {
        public int empId { get; private set; }
        public string empName { get; private set; }
        public string email { get; set; }
        public DateOnly dateOfJoin { get; set; }
        public DateOnly dateOfBirth { get; set; }
        public Employee(int empId, string empName, string email, DateOnly dateOfJoin, DateOnly dateOfBirth)
        {
```

```
            this.empId = empId;
            this.empName = empName;
            this.email = email;
            this.dateOfJoin = dateOfJoin;
            this.dateOfBirth = dateOfBirth;
        }
        public override string ToString()
        {
            return $"{empId} - {empName}";
        }
    }
}
```

Manager.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Employee
{
    internal class Manager:Employee
    {
        private int salary;
        public Manager(int empId, string empName, string email, DateOnly dateOfJoin, DateOnly dateOfBirth,int
salary) : base(empId, empName, email, dateOfJoin, dateOfBirth)
        {
            this.salary = salary;
        }
        public override string ToString()
```

```
        {
            return base.ToString();
        }
    }
}
```

IEmpRepo.cs

```
using System.Collections.Generic;
namespace Employee.Repo
{
    internal interface IEmpRepo<T>
    {
        void Add(T item);
        List<T> GetAll();
    }
}
```

EmpRepo.cs

```
using System.Collections.Generic;
using System.Linq;
namespace Employee.Repo
{
    internal class EmpRepo<T> : IEmpRepo<T>
    {
        private static List<T> items = new List<T>();

        public void Add(T item)
        {
```

```
            items.Add(item);
        }
        public List<T> GetAll()
        {
            return items.ToList();
        }
    }
}
```

output: