**C# Assignment-5**

## **Objective**

* **To use non – generic and generic classes provided in collection framework.**
* **To implement in-built collection interfaces.**
* **To create a custom generic class.**

## **Assignments to be done in this session**

1. **Create following types of arrays**
   1. **Integer**
   2. **String**

**Use System.Array class to perform following operations on them**

**Copy, Sort, Clear, Reverse**

**Accept input from user through Console.**

**Program:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Diagnostic

s;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment

{

class Arrayprogram

{

static void Main(string[] args)

{

Console.Write("Enter size of Array:");

int ArrSize = int.Parse(Console.ReadLine());

int[] arrayInt = new int[ArrSize];

string[] arrayStr = new string[ArrSize];

for (int i = 0; i < ArrSize; i++)

{

Console.Write("Enter {0} int type element in array: ", i);

arrayInt[i] = int.Parse(Console.ReadLine());

}

Console.WriteLine("");

Console.Write("Output before applying any method : ");

foreach (int val in arrayInt)

{

Console.Write(val + " ");

}

Console.WriteLine("\n");

int[] array2 = new int[ArrSize];

Array.Copy(arrayInt, array2, ArrSize);

Console.WriteLine("");

Console.Write("After processing one array in array2 : ");

Console.Write("");

foreach (int value in array2)

{

Console.Write(value + " ");

}

Console.WriteLine("");

Console.WriteLine("Cleared array2: ");

Array.Clear(array2, 0, ArrSize);

foreach (int val in array2)

{

Console.Write(val + " ");

}

Console.WriteLine();

Array.Reverse(arrayInt);

foreach (int val in arrayInt)

{

Console.Write(val + " ");

}

Array.Sort(arrayInt);

Console.WriteLine("");

Console.Write("\nAfter Sorting array: ");

foreach (int val in arrayInt)

{

Console.Write(val + " ");

}

Console.WriteLine("");

Console.WriteLine("-----------------------------------");

Console.WriteLine("\nEnter String in Array\n");

for (int i = 0; i < ArrSize; i++)

{

Console.Write("Enter {0} String type element in array: ", i);

arrayStr[i] = Console.ReadLine();

}

Console.WriteLine("");

Console.Write("Output of Array String before appplying any operation: ");

foreach (string value in arrayStr)

{

Console.Write(value + " ");

}

Array.Reverse(arrayStr);

Console.WriteLine("");

Console.WriteLine("");

Console.Write("After reversing String Array: ");

foreach (string value in arrayStr)

{

Console.Write(value + " ");

}

Console.WriteLine("");

Array.Sort(arrayStr);

Console.Write("Sorting String Array: ");

foreach (string value in arrayStr)

{

Console.Write(value + " ");

}

string[] arraystring2 = new string[ArrSize];

Array.Copy(arrayStr, arraystring2, ArrSize);

Console.WriteLine("");

Console.Write("After copying to Array String2 : ");

foreach (string value in arraystring2)

{

Console.Write(value + " ");

}

Console.WriteLine("");

Console.WriteLine("After Clearing String Array2 : ");

Array.Clear(arraystring2, 0, ArrSize);

foreach (string value in arrayStr)

{

Console.Write(value + " ");

}

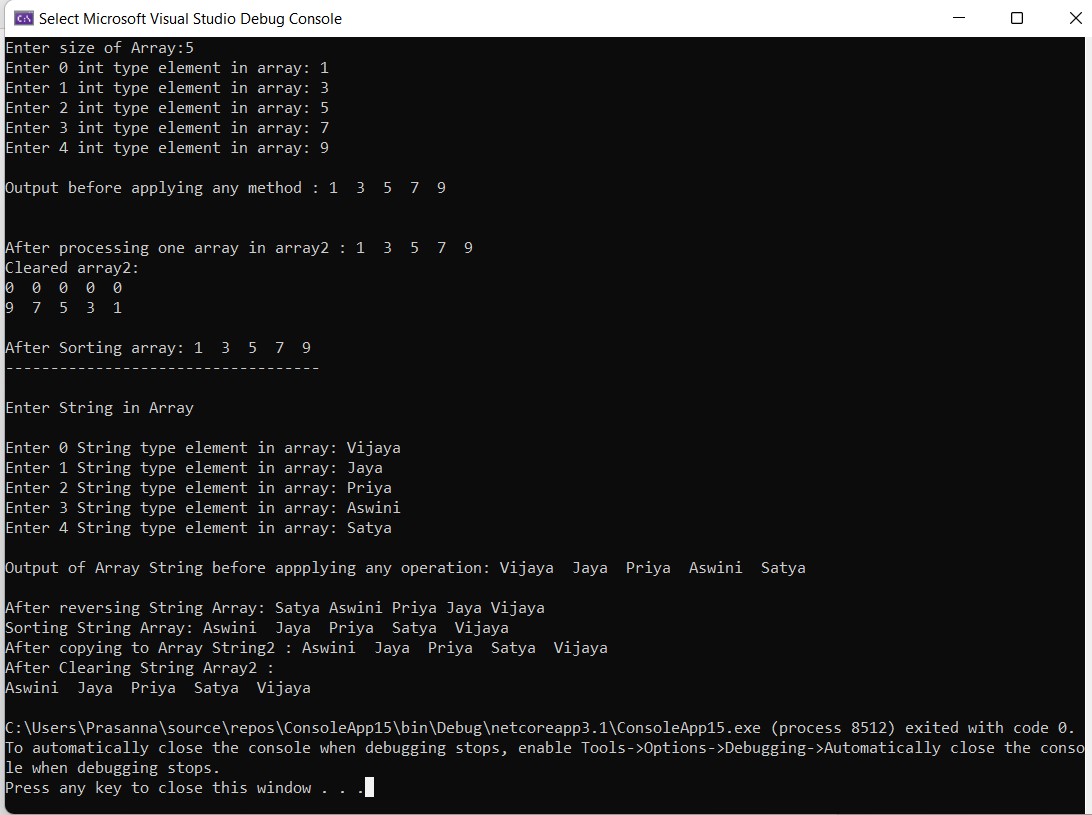
Console.ReadLine();

}

}

}

**OUTPUT:**

****

**2.Use collection class such as ArrayList to hold more than one employee objects in Employee Management application. Display all Employee details which are stored in collection**.

**Program:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

namespace Array

{

public class Employee

{

int EmpId;

string EmpName;

double Empsalary;

public Employee(int Employeeid, string Employeename, double Employeesalary)

{

this.EmpId = Employeeid;

this.EmpName = Employeename;

this.Empsalary = Employeesalary;

}

public override string ToString()

{

return

String.Format("Employee Id: {0} Employee Name: {1} Employee Salary: {2} ", EmpId, EmpName, Empsalary);

}

static void Main(string[] args)

{

ArrayList Employeedetail = new ArrayList();

Employeedetail.Add(new Employee(101, "Vijaya", 100000));

Employeedetail.Add(new Employee(102, "Jaya", 200000));

Employeedetail.Add(new Employee(103, "Satya", 300000));

Employeedetail.Add(new Employee(104, "Durga", 400000));

Employeedetail.Add(new Employee(105, "Aswini", 500000));

Employeedetail.Add(new Employee(106, "Venkat", 600000));

Employeedetail.Add(new Employee(106, "Mallesh", 700000));

Employeedetail.Add(new Employee(106, "Priya", 800000));

Console.WriteLine("Employee Details by using array list");

foreach (Employee Employee in Employeedetail)

{

Console.WriteLine(Employee);

}

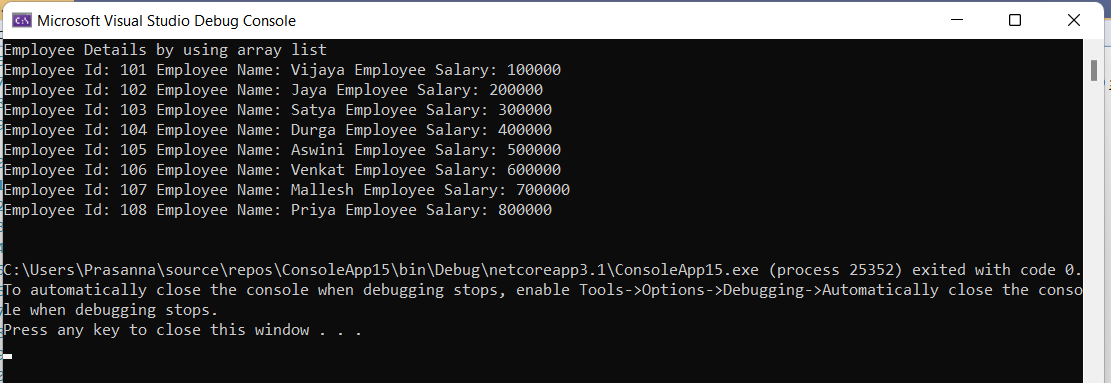
Console.ReadLine();

}

}

}

**OUTPUT:**



**3.Write a console based program to create a linked list of Employee objects using the generic class List<>.Perform following operations on the list:**

**a)Add a new employee**

**b)Display the list of employees.**

**c)Total number of employees in the list**

**Program:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Linq;

namespace LinkedListGenExample

{

public class LinnkedListAssignment

{

static void Main(string[] args)

{

int i, n;

LinkedList<string> NewEmp = new LinkedList<string>();

Console.Write("Enter Number of Employees you want to ADD:");

n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter {0} elements in the list:", n);

for (i = 0; i < n; i++)

{

Console.Write("Enter Employee in the list:", i);

string empList = Console.ReadLine();

NewEmp.AddFirst(empList);

}

Console.WriteLine("List of an Employee:");

foreach (string Employee in NewEmp)

{

Console.WriteLine(Employee);

}

Console.WriteLine("Total number of Employee : " + NewEmp.Count());

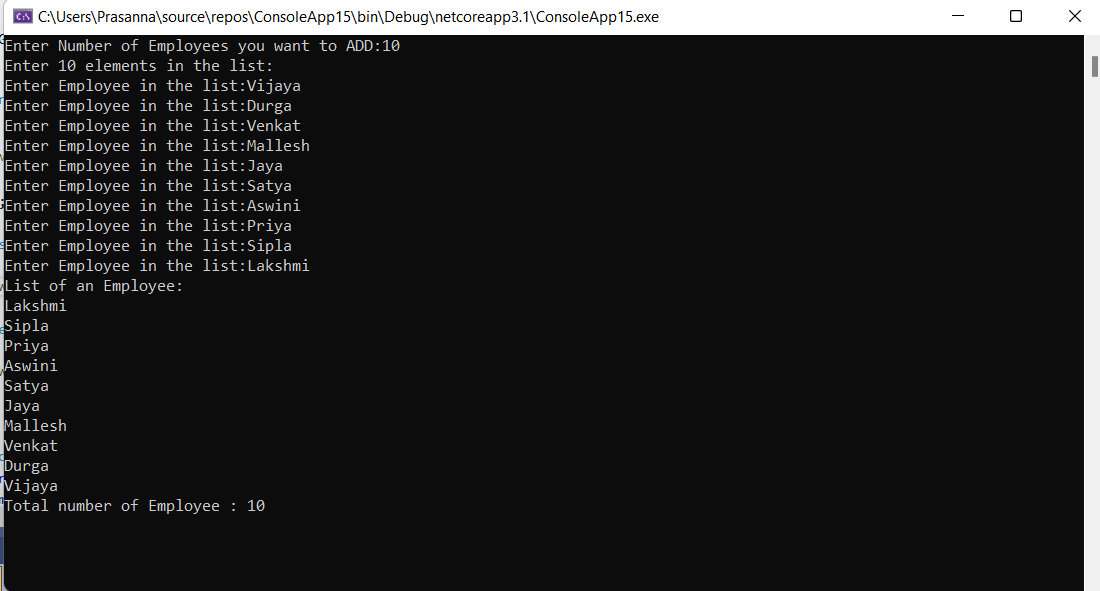
Console.ReadLine();

}

}

}

**Output:**

****

**4.Write Custom Generic class MyStack based on assignment of previous**

**session, with**

**Push() and Pop() methods to store any kind of .NET Type.**

**Program:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Linq;

namespace Stack

{

class MyStack

{

static void Main(string[] args)

{

Stack<Object> genstack = new Stack<Object>();

Console.WriteLine("Apply Element in Stack");

genstack.Push(85);

genstack.Push("Vijaya");

genstack.Push("Jaya");

genstack.Push(33.44444);

foreach (Object ob in genstack)

{

Console.WriteLine(ob);

}

Console.WriteLine("");

Console.WriteLine("Pop Operation is Applied on {0}", genstack.Pop());

Console.WriteLine("After pop() operation");

foreach (Object Object in genstack)

{

Console.WriteLine(Object);

}

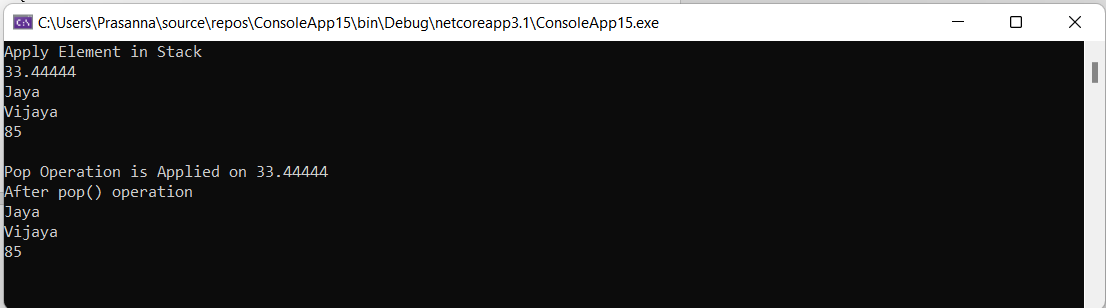
Console.ReadLine();

}

}

}

**Output:**

****

## **Now try these to get a complete grip…**

**5.In the assignment 3 above, add a functionality to search an employee on name in the List<>.**

**Program:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Data;

namespace LinkedListGenExample

{

public class LinnkedListAssignment

{

static void Main(string[] args)

{

int i, n;

LinkedList<string> NewEmp = new LinkedList<string>();

Console.Write("Enter Number of Employee you want to ADD:");

n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter {0} elements in the list:", n);

for (i = 0; i < n; i++)

{

Console.Write("Enter Employee in the list:", i);

string empList = Console.ReadLine();

NewEmp.AddFirst(empList);

}

Console.WriteLine("List of an Employee:");

foreach (string employee in NewEmp)

{

Console.WriteLine(employee);

}

Console.WriteLine("Total number of Employee : " + NewEmp.Count());

Console.Write("Enter the of Employee you want to Search: ");

string search = Console.ReadLine();

String res = Convert.ToString(NewEmp.Contains(search));

Console.WriteLine("Contains Employee in List: " + res

Console.ReadLine();

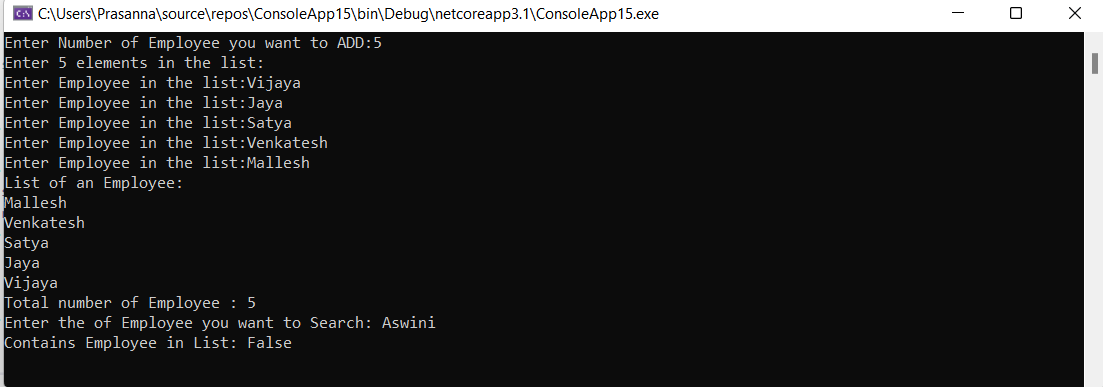
}

}

}

**OUTPUT:**

****

****