Stack:

Array list is represent sequential data

Stack is represent data in a form of LIFO last in first out

Ex – stack 1 = [a, b, c, d]

LIFO = d c b a

Insert element in a stack using push method.

Remove element in a stack using pop method. Pop is used to remove only one element at a time.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace stack\_day12

{

class Program

{

static void Main(string[] args)

{

Stack s1 = new Stack();

//insert element in a stack

s1.Push('A');

s1.Push('B');

s1.Push('C');

s1.Push('D');

s1.Push('E');

Console.Write("Element in stack : \t");

foreach(char i in s1)

{

Console.Write(i + "\t");

}

//Remove stack elements

s1.Pop();

Console.Write("\nAfter remove elements : \t");

foreach (char i in s1)

{

Console.Write(i + "\t");

}

s1.Pop();

Console.Write("\nAfter remove elements : \t");

foreach (char i in s1)

{

Console.Write(i + "\t");

}

Console.ReadKey();

}

}

}

Task: Assignment: 1

Create n number of stack.

Input all the element for the user inside the stack.

Input choice form the user for removing data elements.

Result:

Enter number of elements: 5

Enter element 1 – A

Enter element 2 - B

Enter element 3 - C

Number of element is to deleted – 2

Hash Table:

Hash table store data elements in a particular key.

Ex - 101 – Nagpur

102 – Mumbai

103 – Pune

Hash table is used to display data information in a random format.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace hash\_1

{

class Program

{

static void Main(string[] args)

{

Hashtable h1 = new Hashtable();

//insert data element in data table

h1.Add("101","Nagpur");

h1.Add("102", "Pune");

h1.Add("103", "Mumbai");

Console.WriteLine("Display HashTable : ");

foreach(var i in h1.Keys)

{

Console.WriteLine(i+" "+h1[i]);

}

Console.ReadLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace hash\_1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter number of hashtable : ");

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

Hashtable h1 = new Hashtable(n);

Console.WriteLine("Enter key and value : ");

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

{

h1.Add(Console.ReadLine(), Console.ReadLine());

}

Console.WriteLine("Display Hash Table : ");

foreach(var i in h1.Keys)

{

Console.WriteLine(i+" "+h1[i]);

}

Console.ReadLine();

}

}

}

Sorted List:

1. Sorted list is used to input all data in a particular key.
2. Sorted list is return all elements in a sorted order.
3. Sorted list is used to create pair object.

Types of object:

1. Single
2. Pair

Pair Object:

Pair object used to var variable.

Class\_name <int, string> o\_name = new class\_name<int, string>();

A<int, string> a1 = new A<int, string>();

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace Sort\_list

{

class Program

{

static void Main(string[] args)

{

//create sorted list object(pair object)

SortedList<int, string> o1 = new SortedList<int, string>();

//insert elements in a sorted list

o1.Add(3, "Nagpur");

o1.Add(2, "Pune");

o1.Add(1, "Mumbai");

o1.Add(6, "Delhi");

//display soretd list data elements

foreach(var a in o1.Keys)

{

Console.WriteLine(a+" "+o1[a]);

}

Console.ReadKey();

}

}

}

Assignment 2:

Combine all program

Arraylist

Stack

Sorted

Hashtable

In a single program

Constructor:

1. Constructor is object initialization process.
2. It is a method similar to call name.

Class A {Public void A () {} }

1. Constructor cannot used void keyword.
2. Constructor method automatically create at the time of object creation.

Type of Constructor

* 1. Default Constructor
  2. Parametrized
  3. With default argument

Default

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Constructor\_1

{

class student

{

public student()

{

Console.WriteLine("This is contructor : ");

}

}

class Program

{

static void Main(string[] args)

{

student s1 = new student();

Console.ReadKey();

}

}

}

Parametrized Constructor

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Constructor\_1

{

class student

{

public student(int a)

{

Console.WriteLine("This is contructor : "+a);

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter Number : ");

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

student s1 = new student(n);

Console.ReadKey();

}

}

}

Assignment 3:

Create a program constructor with inheritance.

Convert any one function assignment into constructor.