**HND Software Development – Data Structure**

**Stack ADT – Using Array Data Structure**

Study the example of Abstract Data Type (ADT) implementation in C# using array data structure and carry out the following;

1. Attempt to setup the implementation in Visual Studio and make the following changes to contain your initials\_name of class (e.g. OE\_Stack\_Interface). Make changes to the followingclasses;

(i) namespace

(ii) ADT Interface

(iii) Array Data structure

(iv) Double the Stack size

(v)

Note: Name this file as *Your Initials\_Array\_ADT\_Example1*

2. Try to comment your codes as much as you can to include your notes or description of any segment of the codes.

3. Attempt to setup the implementation differently by having all the classes combined in a single code file.

**Note**: Name this file as *Your Initials\_Array\_ADT\_Example1*

4. Carry out research on linked list ADT and attempt to setup implementation in questions 1 – 3 above. Note: you will need to replace ‘Array’ with ‘LinkedList’ in your classes.

**Part A** – Setting up Array – Stack ADT using different classes.

**Step 1**. -

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DS\_Stack\_ADT\_Array

{

class Program

{

static void Main(string[] args)

{

stack st = new stack();

while (true)

{

Console.Clear();

Console.WriteLine("\nStack MENU(size -- 10)");

Console.WriteLine("1. Add an element");

Console.WriteLine("2. See the Top element.");

Console.WriteLine("3. Remove top element.");

Console.WriteLine("4. Display stack elements.");

Console.WriteLine("5. Exit");

Console.Write("Select your choice: ");

int choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine("Enter an Element : ");

st.Push(Console.ReadLine());

break;

case 2: Console.WriteLine("Top element is: {0}", st.Peek());

break;

case 3: Console.WriteLine("Element removed: {0}", st.Pop());

break;

case 4: st.Display();

break;

case 5: System.Environment.Exit(1);

break;

}

Console.ReadKey();

}

}

}

**Step 2**. -

**Add an ADT Interface – Interface StackADT**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DS\_Stack\_ADT\_Array

{

class Program

{

interface StackADT

{

Boolean isEmpty();

void Push(Object element);

Object Pop();

Object Peek();

void Display();

}

}

}

**Step 3**. -

**Add a Class – StackADT**

**Note**: The class ***StackADT*** added will which will provide an array data structure to which integer items can be added and removed on a FILO basis

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DS\_Stack\_ADT\_Array

{

class stack : StackADT

{

private int StackSize;

public int StackSizeSet

{

get { return StackSize; }

set { StackSize = value; }

}

public int top;

Object[] item;

public stack()

{

StackSizeSet = 10;

item = new Object[StackSizeSet];

top = -1;

}

public stack(int capacity)

{

StackSizeSet = capacity;

item = new Object[StackSizeSet];

top = -1;

}

public bool isEmpty()

{

if (top == -1) return true;

return false;

}

public void Push(object element)

{

if (top == (StackSize - 1))

{

Console.WriteLine("Stack is full!");

}

else

{

item[++top] = element;

Console.WriteLine("Item pushed successfully!");

}

}

public object Pop()

{

if (isEmpty())

{

Console.WriteLine("Stack is empty!");

return "No elements";

}

else

{

return item[top--];

}

}

public object Peek()

{

if (isEmpty())

{

Console.WriteLine("Stack is empty!");

return "No elements";

}

else

{

return item[top];

}

}

public void Display()

{

for (int i = top; i > -1; i--)

{

Console.WriteLine("Item {0}: {1}", (i + 1), item[i]);

}

}

}

}