**Task-1**

**Queue**

**Definition:**

**Queue** represents a ***first-in, first out*** collection of objects. It is used when you need a first-in, first-out access of items.

**Features:**

* Enqueue adds an element to the end of the queue
* Dequeue removes the oldest element from the start of the queue
* Peek returns the oldest element that is at the start of the queue but does not remove it.
* The capacity of the queue is the number of elements it can hold.
* Queue accepts NULL as a valid value and allows duplicate elements.

**Questions:**

* Declaration of queue?

Ans: Queue mqQueue = new Queue();

* Queue collection is in which namespace?

Ans: System.Collections;

* The property that returns the total number of elements in queue

Ans: Count property | Example: myQueue.Count();

* Getting the first element without removing the element

Ans: Peek();

* Removing all the elements in the queue

Ans: Clear();

**Exercise 1:**

using System;

using System.Collections;

class GFG

{

public static void Main()

{

Queue myQueue = new Queue();

myQueue.Enqueue("Chandigarh");

myQueue.Enqueue("Delhi");

myQueue.Enqueue("Noida");

myQueue.Enqueue("Himachal");

myQueue.Enqueue("Punjab");

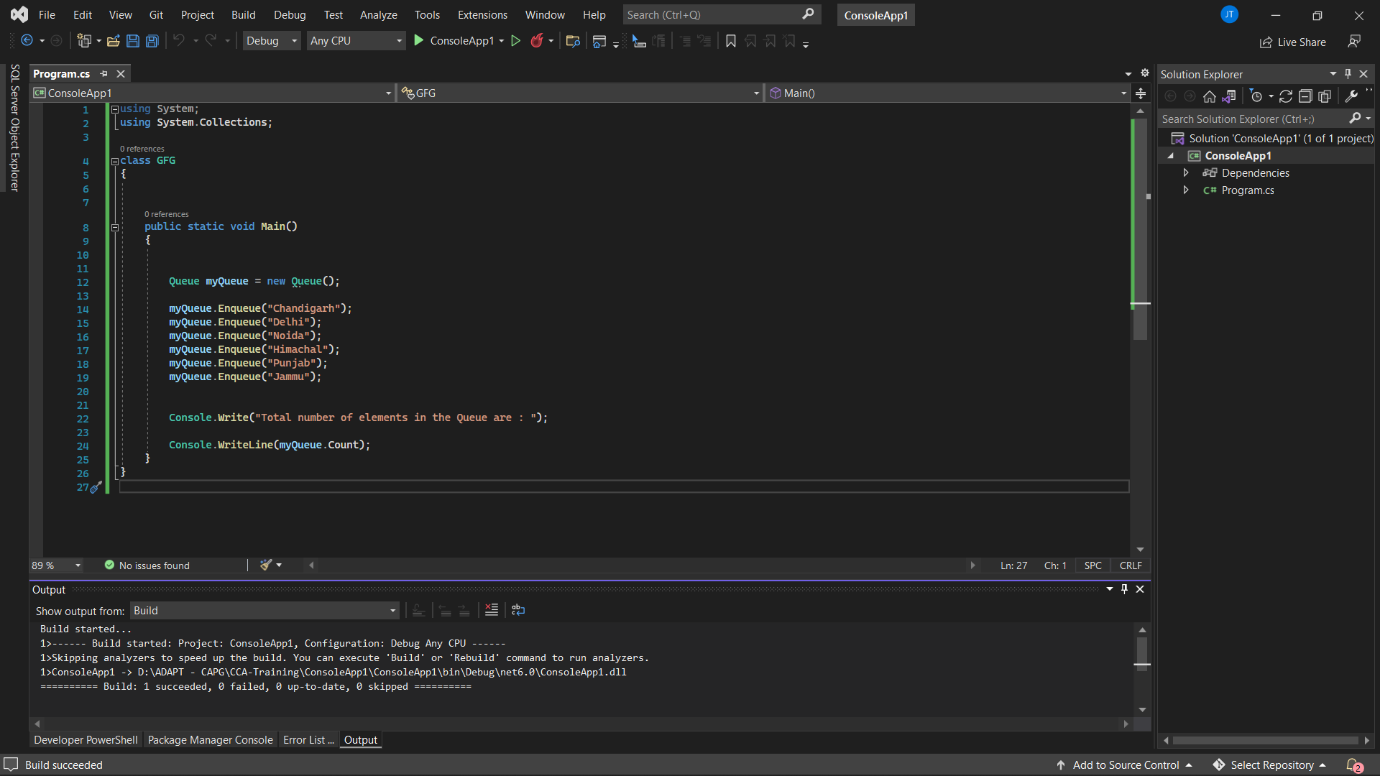
myQueue.Enqueue("Jammu");

Console.Write("Total number of elements in the Queue are : ");

Console.WriteLine(myQueue.Count);

}

}



**Exercise 2:**

using System;

using System.Collections;

class GFG

{

public static void Main()

{

Queue myQueue = new Queue();

myQueue.Enqueue("Chandigarh");

myQueue.Enqueue("Delhi");

myQueue.Enqueue("Noida");

myQueue.Enqueue("Himachal");

myQueue.Enqueue("Punjab");

myQueue.Enqueue("Jammu");

var obj = myQueue.Peek();

myQueue.Enqueue(obj);

foreach(var ob in myQueue)

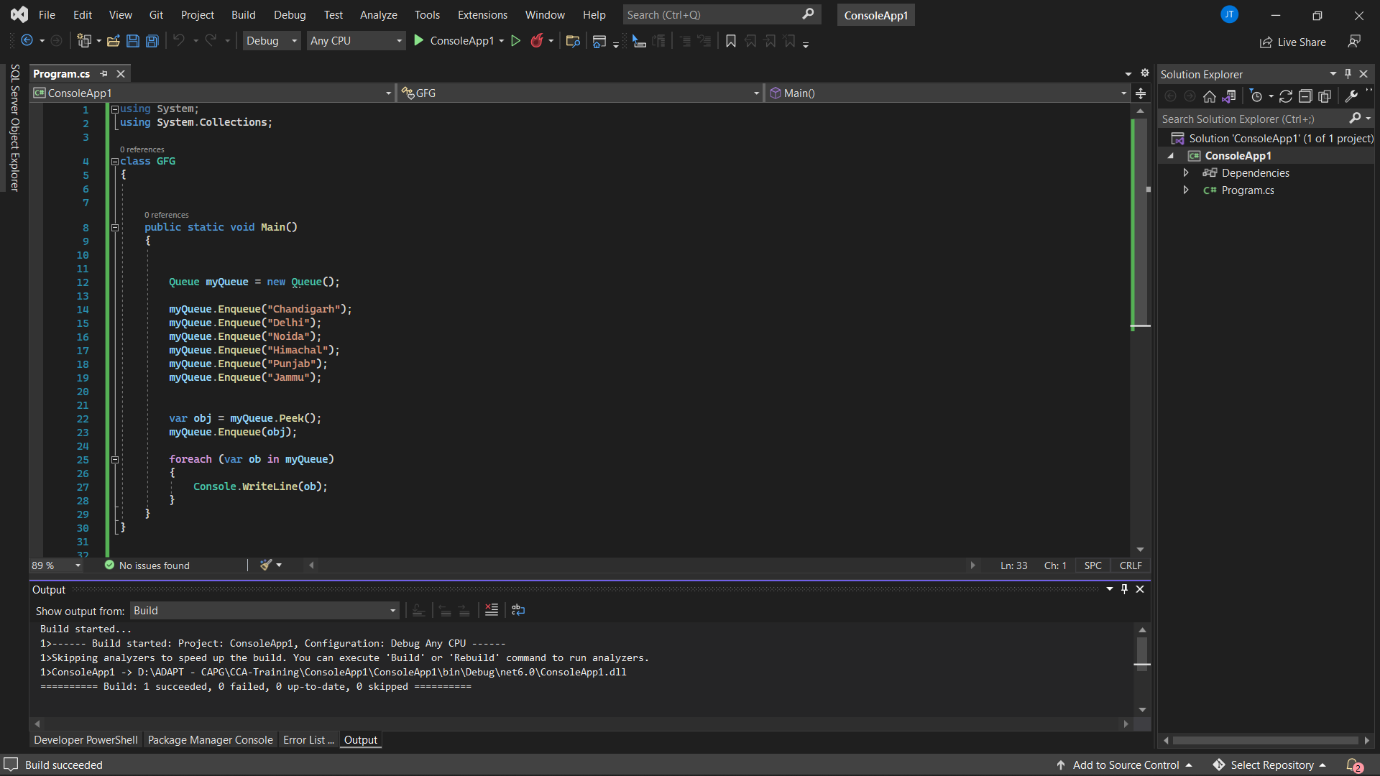
{

Console.WriteLine(ob);

}

}

}



**Exercise 3:**

using System;

using System.Collections;

class GFG

{

public static void Main()

{

Queue myQueue = new Queue();

myQueue.Enqueue("Chandigarh");

myQueue.Enqueue("Delhi");

myQueue.Enqueue("Noida");

myQueue.Enqueue("Himachal");

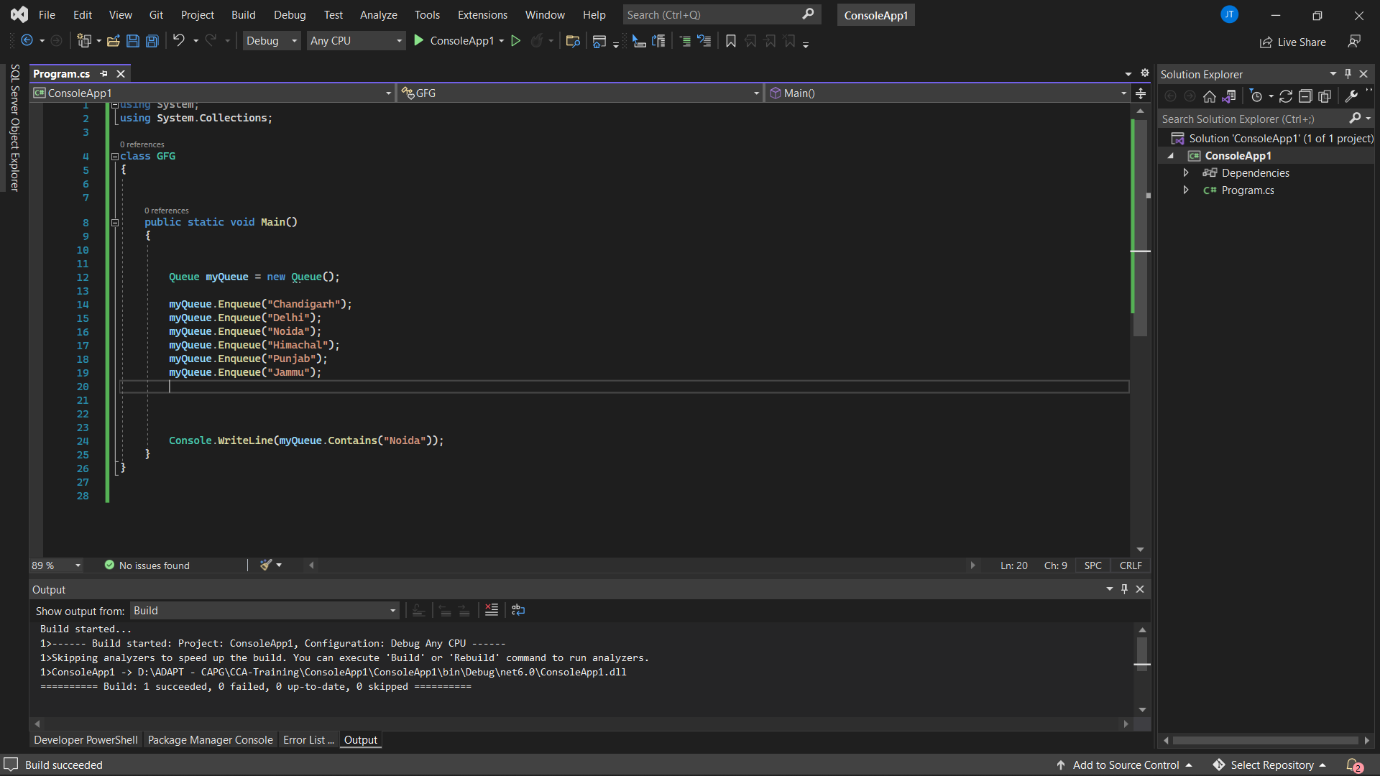
myQueue.Enqueue("Punjab");

myQueue.Enqueue("Jammu");

Console.WriteLine(myQueue.Contains("Noida"));

}

}

****

**Exercise 4:**

using System;

using System.Collections;

class GFG

{

public static void Main()

{

Queue myQueue = new Queue();

myQueue.Enqueue("Chandigarh");

myQueue.Enqueue("Delhi");

myQueue.Enqueue("Noida");

myQueue.Enqueue("Himachal");

myQueue.Enqueue("Punjab");

myQueue.Enqueue("Jammu");

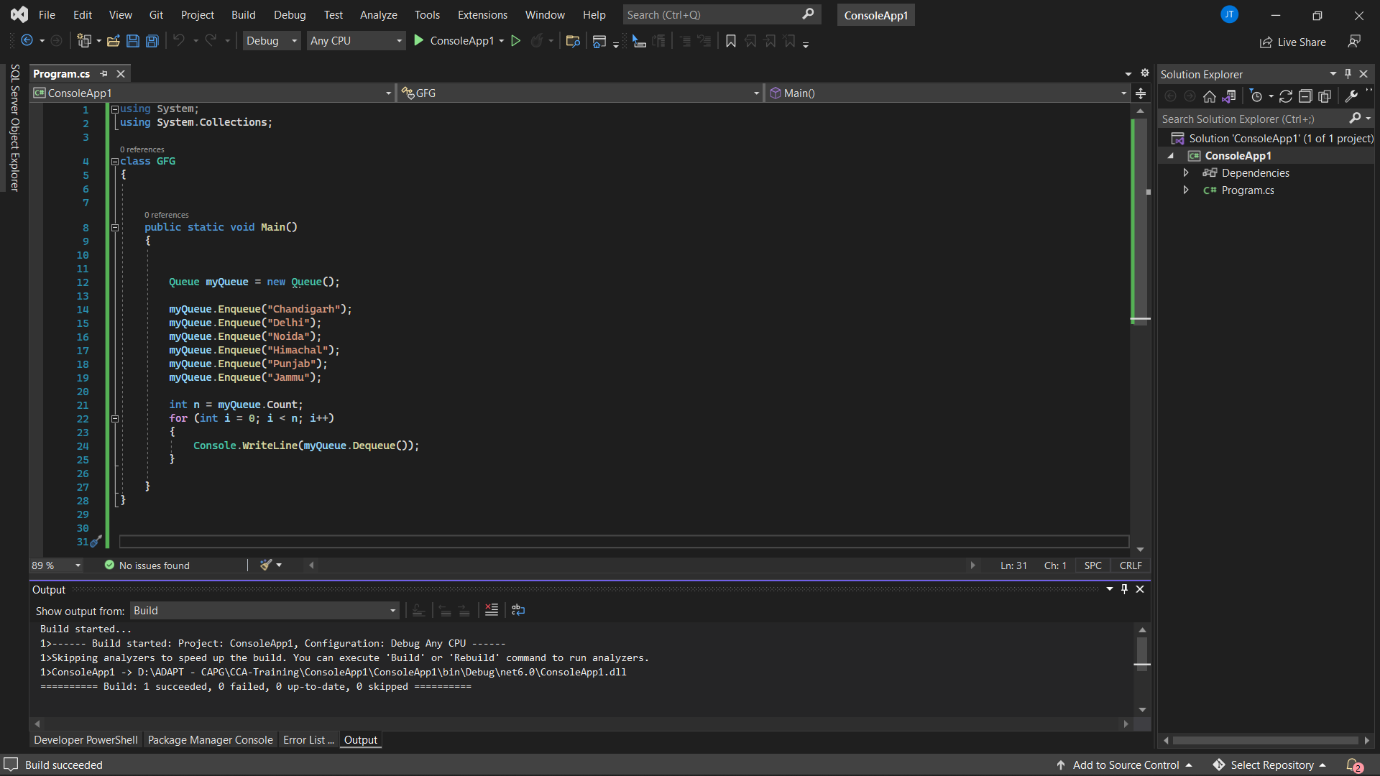
int n = myQueue.Count;

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

{

Console.WriteLine(myQueue.Dequeue());

}}}



**Exercise 5:**

using System;

using System.Collections;

class GFG

{

public static void Main()

{

Queue myQueue = new Queue();

myQueue.Enqueue("Chandigarh");

myQueue.Enqueue("Delhi");

myQueue.Enqueue("Noida");

myQueue.Enqueue("Himachal");

myQueue.Enqueue("Punjab");

myQueue.Enqueue("Jammu");

Stack myStack = new Stack();

while(myQueue.Count > 0)

{

myStack.Push(myQueue.Dequeue());

}

while (myStack.Count > 0)

{

myQueue.Enqueue(myStack.Pop());

}

foreach (var ob in myQueue)

{

Console.WriteLine(ob);

}

}

}

