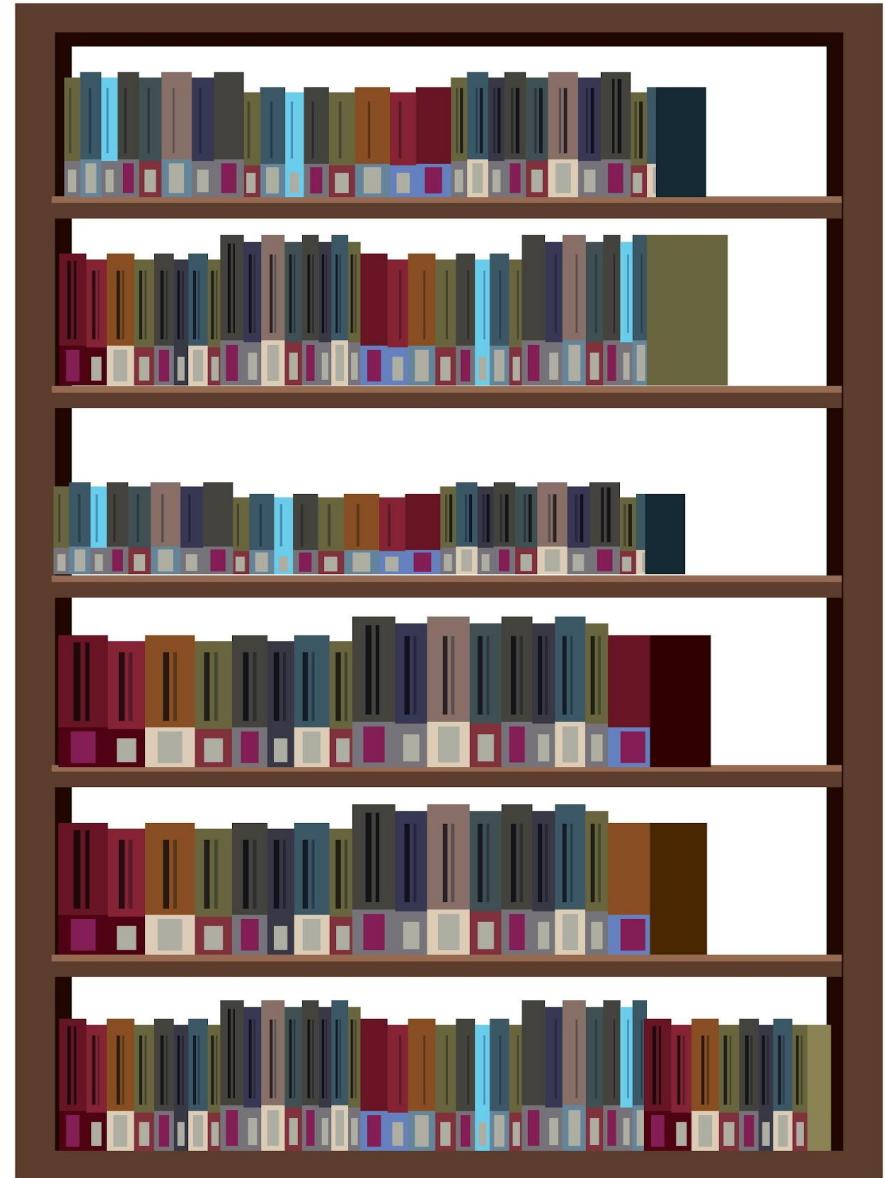
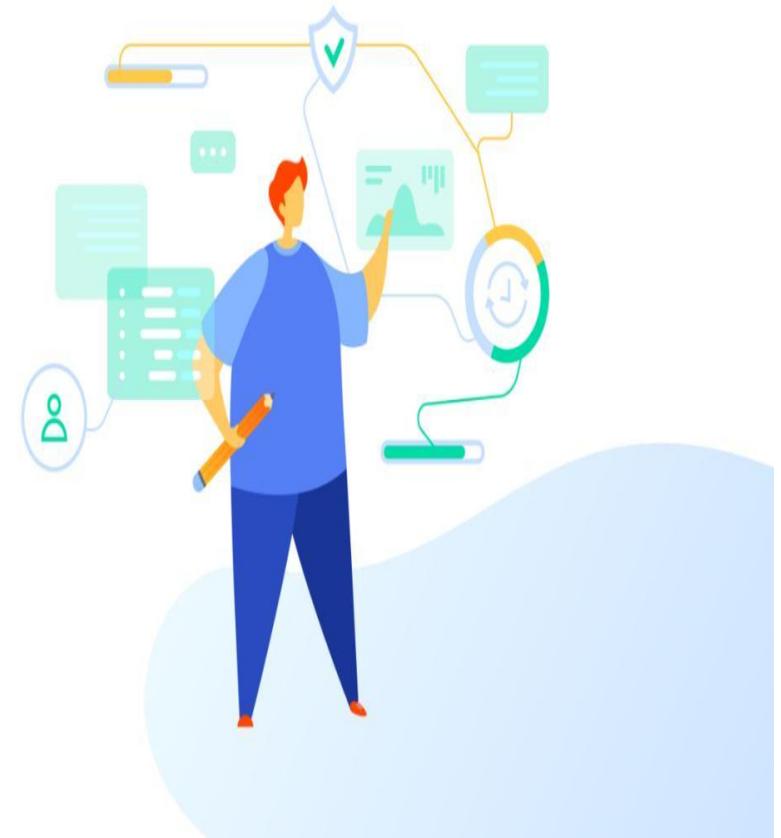


COLLECTIONS



Overview



INTRODUCTION

A collection sometimes called a container is simply an object that groups multiple elements into a single unit.

A collection is a kind of data structure.

A data structure is a way of storing data in a computer so that it can be used efficiently.

Collections can improve code readability and self documentation, as well as enhance maintainability.

C# Collection classes are defined as part of the System.Collections



TYPES OF COLLECTIONS IN .NET

Non Generic

- ArrayList
- Hashtable
- SortedList
- Queue
- Stack

Generics

- List<T>
- Dictionary< TKey, TValue >
- SortedList< TKey, TValue >
- Queue< T >
- Stack< T >

NON-GENERIC COLLECTIONS



Non-generic collections in C# are collections that can hold objects of any type.

They are defined in the `System.Collections` namespace and include the following classes:



ArrayList



SortedList



HashTable



Stack



Queue

Dynamic array implementation that can store objects of any type

Collection class that represents a collection of key-value pairs sorted by key.

It represents a collection of key-value pairs in which keys are unique and the values can be of any type

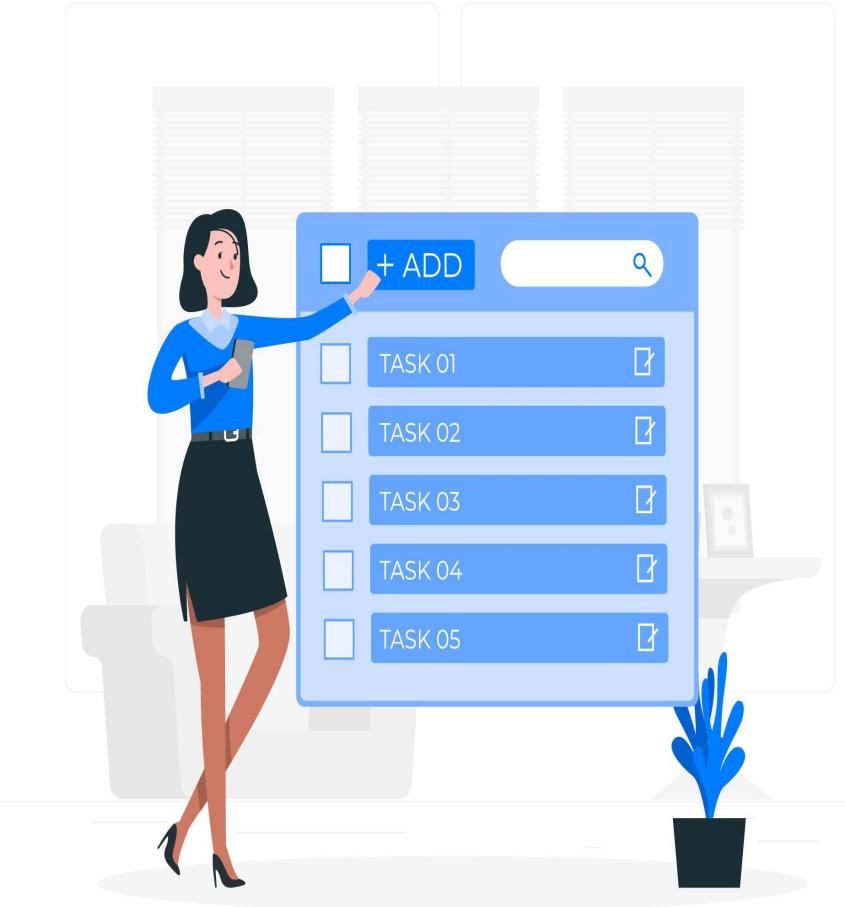
Stack is a collection class that represents a last-in, first-out (LIFO) collection of objects.

Queue is a collection class that represents a first-in, first-out (FIFO) collection of objects.

ARRAYLIST

In C#, an ArrayList is a dynamic array implementation that can store objects of any type.

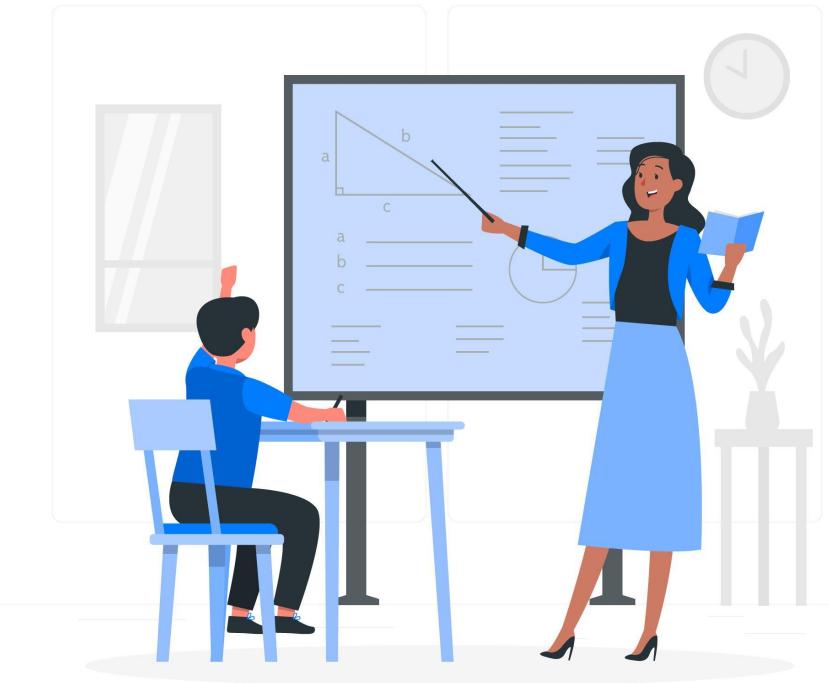
It provides a resizable array-like data structure with automatic expansion and contraction capabilities.



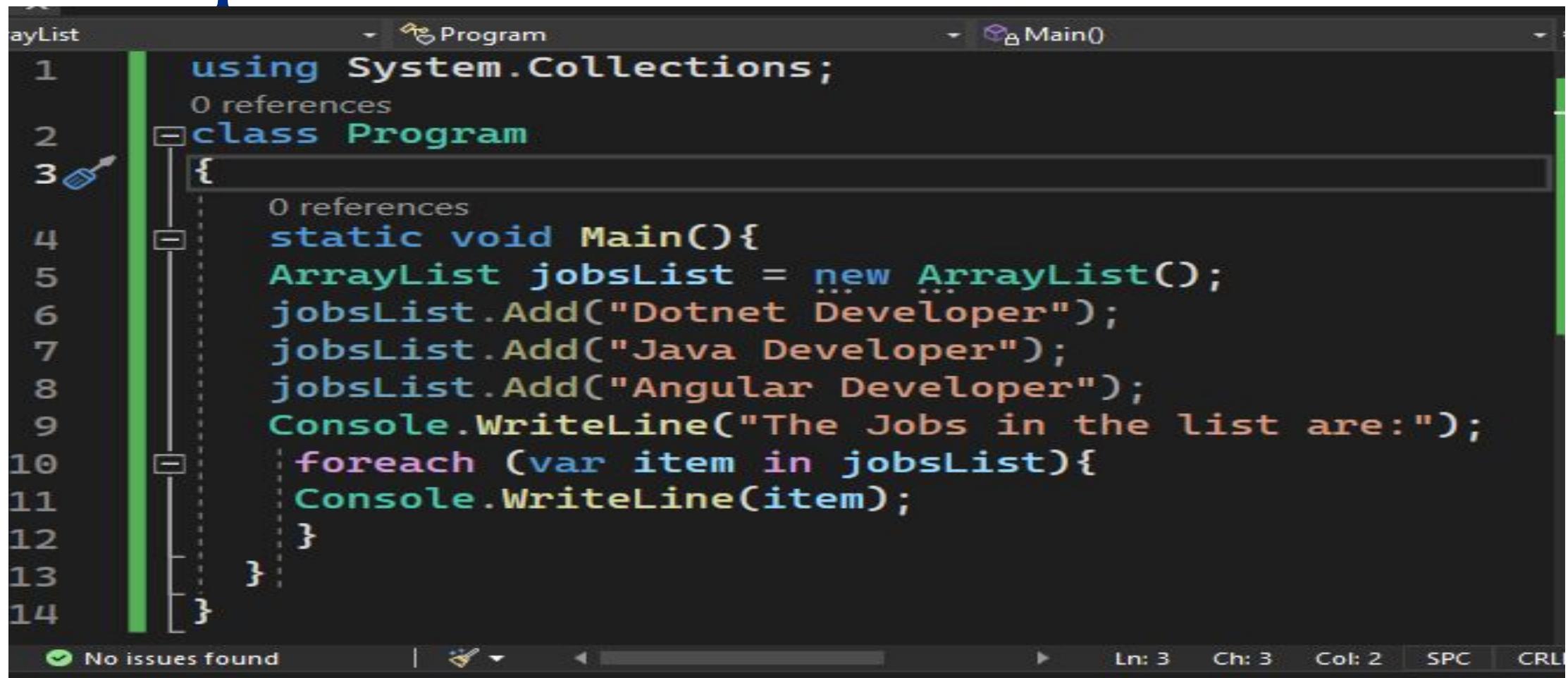
The ArrayList class is defined in the System.Collections namespace

It provides methods such as

- Add
- Remove
- RemoveAt
- Insert
- IndexOf
- Contains
- Count
- Sort



Example



A screenshot of a code editor window titled "Program" showing a C# program. The code uses an `ArrayList` to store job titles and prints them to the console.

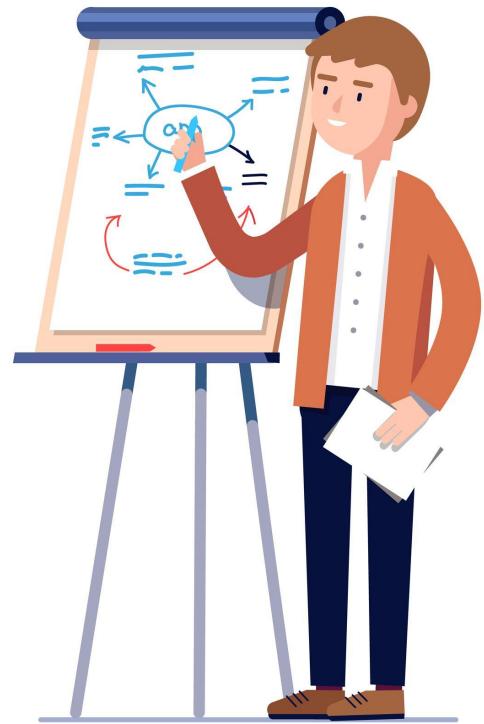
```
ayList
  Program
    Main()
1  using System.Collections;
2  class Program
3  {
4      static void Main(){
5          ArrayList jobsList = new ArrayList();
6          jobsList.Add("Dotnet Developer");
7          jobsList.Add("Java Developer");
8          jobsList.Add("Angular Developer");
9          Console.WriteLine("The Jobs in the list are:");
10         foreach (var item in jobsList){
11             Console.WriteLine(item);
12         }
13     }
14 }
```

The code editor interface includes a vertical navigation bar on the left, a status bar at the bottom with "No issues found", and a toolbar with various icons.

HASHTABLE

In C#, a Hashtable is a collection class that represents a collection of key-value pairs in which keys are unique and the values can be of any type.

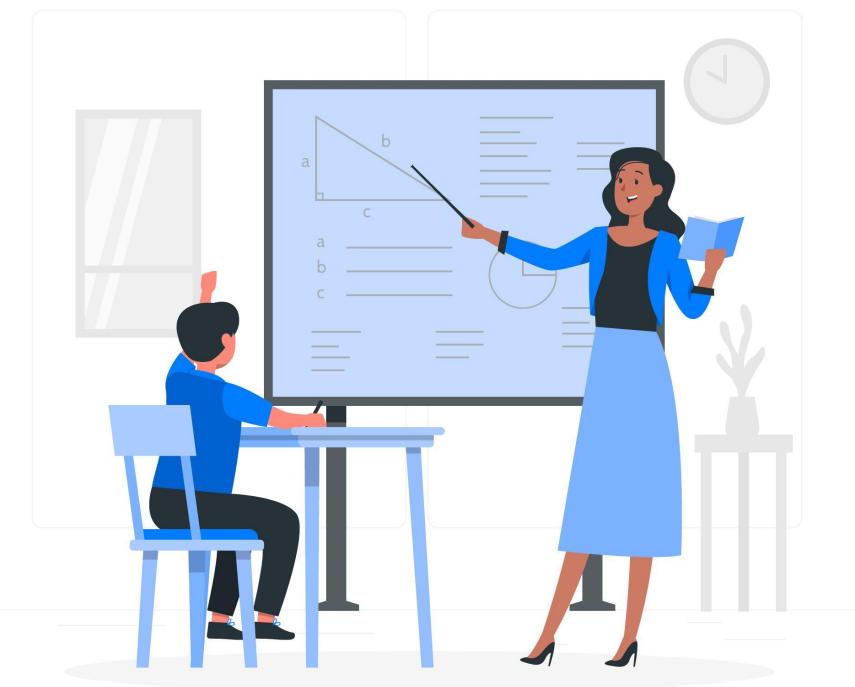
It provides a fast way to store and retrieve data by using a hash table algorithm that maps keys to their corresponding values.



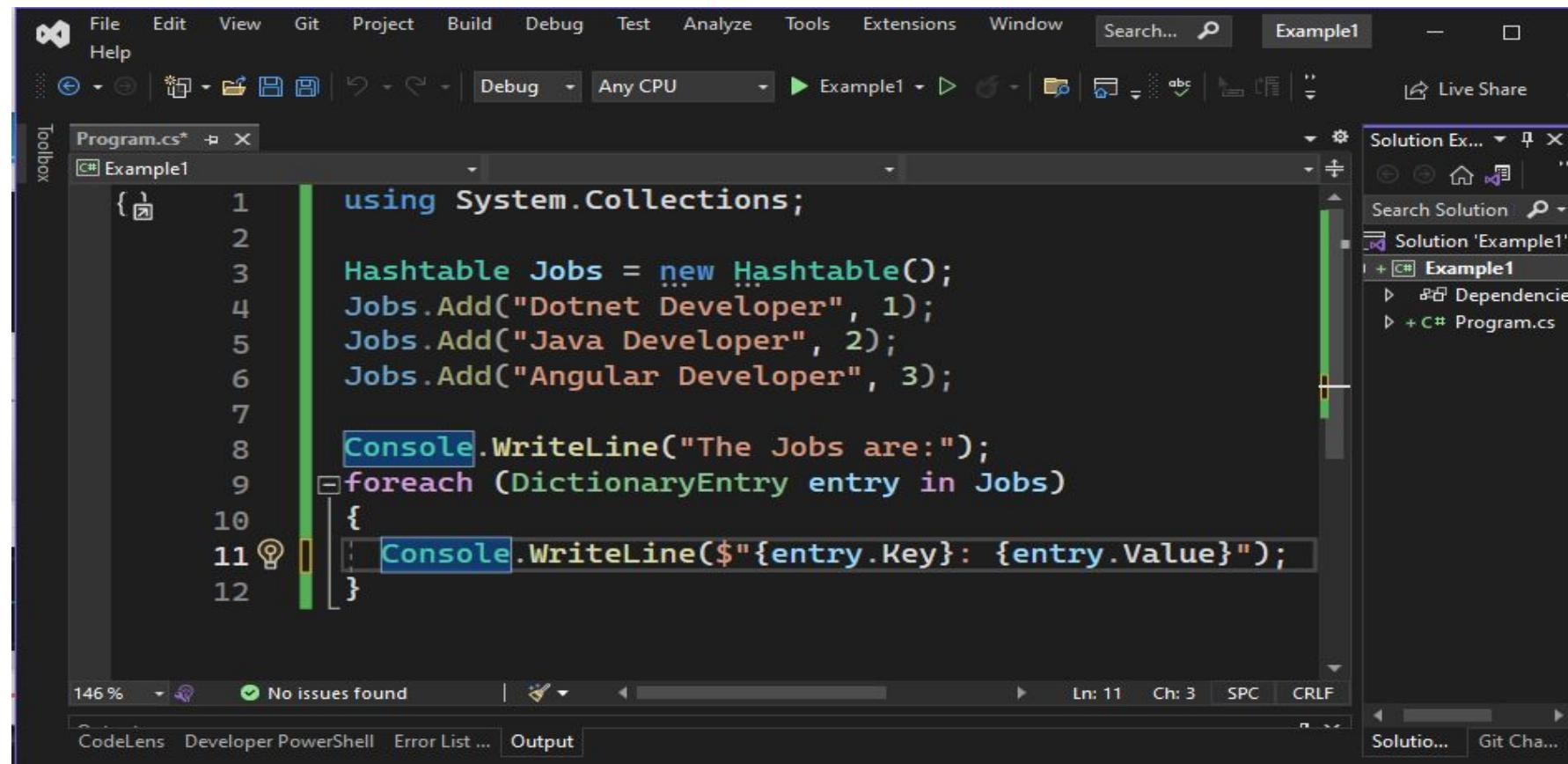
The Hashtable class is defined in the System.Collections namespace

It provides methods such as

- Add
- Remove
- Contains
- ContainsKey
- ContainsValue
- Count.



Example



The screenshot shows a Microsoft Visual Studio interface with a dark theme. The main window displays a C# code editor for a file named "Program.cs". The code uses a HashTable to store developer names and their counts, then iterates through the entries to print them to the console.

```
using System.Collections;

Hashtable Jobs = new Hashtable();
Jobs.Add("Dotnet Developer", 1);
Jobs.Add("Java Developer", 2);
Jobs.Add("Angular Developer", 3);

Console.WriteLine("The Jobs are:");
foreach (DictionaryEntry entry in Jobs)
{
    Console.WriteLine($"{entry.Key}: {entry.Value}");
}
```

The Solution Explorer on the right shows a single solution named "Example1" containing a project named "Example1" with a "Program.cs" file. The status bar at the bottom indicates "No issues found".

QUEUE

In C#, a Queue is a collection class that represents a first-in, first-out (FIFO) collection of objects.

It provides a way to store and retrieve objects in a "first in, first out" order, where the first object added is the first object to be removed.



The Queue class is defined in the `System.Collections` namespace

It provides methods such as

- Enqueue
- Dequeue,
- Peek
- Count.



Example

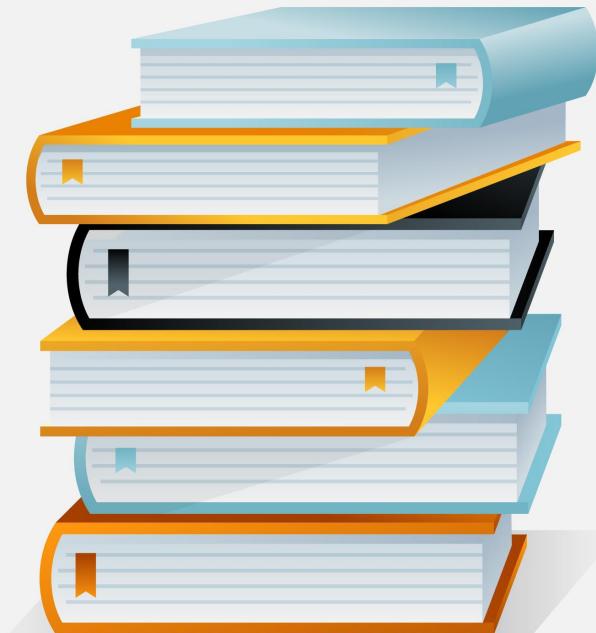
```
using System.Collections;
0 references
class Program
{
    0 references
    static void Main()
    {
        //creating a queue
        Queue jobs = new Queue();
        //Adding members to the Queue
        jobs.Enqueue("Dotnet Developer");
        jobs.Enqueue("Java Developer");
        jobs.Enqueue("Angular Developer");

        Console.WriteLine("The Jobs in the queue are:");
        while (jobs.Count > 0)
        {
            //Removes And Returns the element at the begining of the queue
            Console.WriteLine(jobs.Dequeue());
        }
    }
}
```

STACK

In C#, a Stack is a collection class that represents a last-in, first-out (LIFO) collection of objects.

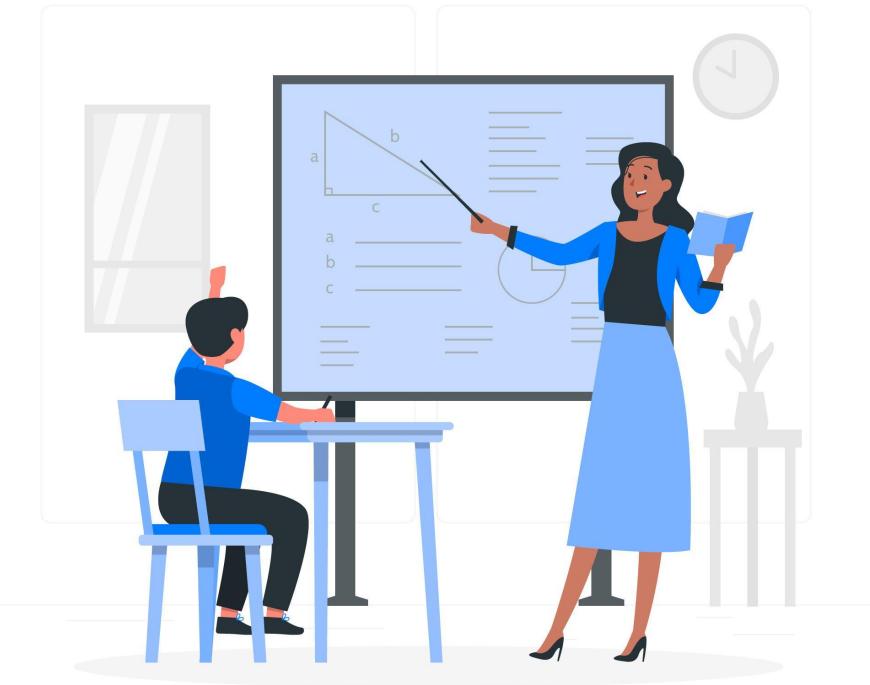
It provides a way to store and retrieve objects in a "Last in, first out" order, where the first object added is the first object to be removed.



The Stack class is defined in the **System.Collections** namespace

It provides methods such as

- Push
- Pop,
- Peek
- Count.
- Contains



Example

```
using System.Collections;
0 references
class Program
{
    0 references
    static void Main()
    {
        //creating a stack
        Stack jobs = new Stack();
        //Adding members to the stack
        jobs.Push("Dotnet Developer");
        jobs.Push("Java Developer");
        jobs.Push("Angular Developer");

        Console.WriteLine("The Jobs in the Stack are:");
        while (jobs.Count > 0)
        {
            //Removes And Returns the element at the begining of the stack
            Console.WriteLine(jobs.Pop());
        }
    }
}
```

SORTED LIST

Sorted List is a collection class that represents a collection of key-value pairs sorted by key.

A key cannot be a null reference, but a value can be.

SortedList is a combination of the Array and the Hashtable.

A SortedList object automatically sort the items in alphabetic or numeric order.

It uses a binary search algorithm to locate elements quickly.



The Sorted List class is defined in the **System.Collections** namespace

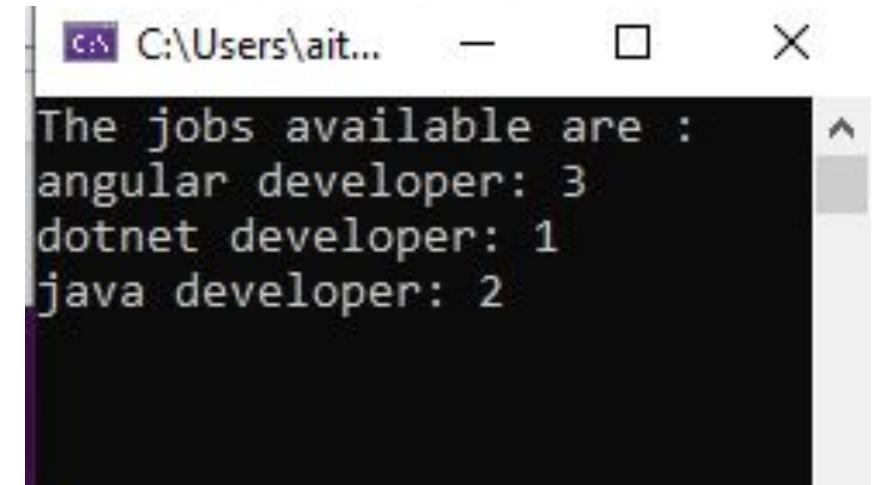
It provides methods such as

- Add
- Remove,
- Contains key
- IndexOfKey
- IndexOfValue
- Count



Example

```
1  using System.Collections;
2  class Program
3  {
4      static void Main()
5      {
6          SortedList jobs = new SortedList();
7          jobs.Add("dotnet developer", 1);
8          jobs.Add("java developer", 2);
9          jobs.Add("angular developer", 3);
10
11         Console.WriteLine("The jobs available are :");
12         foreach (var key in jobs.Keys)
13         {
14             Console.WriteLine($"{key}: {jobs[key]}");
15         }
16     }
17 }
18
```



```
C:\Users\ait... - □ X
The jobs available are :
angular developer: 3
dotnet developer: 1
java developer: 2
```



thank
you