PATRICK CHIAMAKA ESTHER

2021513103

CSC 252

ASSIGNMENT

1. Declaring and initializing an array in C#:

Declaring and initializing an array in C# involves specifying the data type and name of the array, as well as its size or capacity.

```csharp
dataType[] arrayName; // Declare the array
arrayName = new dataType[arraySize]; // Initialize the array with a specified size
```

To declare and initialize an integer array with a size of 5:

```csharp
int[] myArray; // Declare the array
myArray = new int[5]; // Initialize the array with a size of 5
```

2. Initializing and declaring a two-dimensional array in C#:

Initializing and declaring a two-dimensional array in C# involves specifying the data type and name of the array, as well as its size and capacity.

```csharp
dataType[,] arrayName; // Declare the array
arrayName = new dataType[rowSize, columnSize]; // Initialize the array with specified sizes
```

~~~

To declare and initialize a two-dimensional integer array with sizes of 3x4:

```csharp

int[,] myArray; // Declare the array

myArray = new int[3, 4]; // Initialize the array with sizes of 3x4 , , ,

3. Transversing an array in C#:

Transversing an array in C# involves iterating through its elements using a loop. There are two common ways to traverse an array: using a traditional for loop or using the for each loop.

```csharp

int[] myArray = { 1, 2, 3, 4, 5 };

for (int i = 0; i < myArray.Length; i++){

    int element = myArray[i];

    Console.WriteLine(element); } , , ,

```csharp

int[] myArray = { 1, 2, 3, 4, 5 };

foreach (int element in myArray) {

    Console.WriteLine(element); { , , ,

4. Concatenating an array on C#:

Concatenating an array in C# involves combining two or more arrays into a single arra y. To concatenate arrays in C#, you can use the Concat() method from the LINQ (Langu age Integrated Query) library.

```csharp

int[] array1 = { 1, 2, 3 };
~~~

```
int[] array2 = { 4, 5, 6 };

int[] concatenatedArray = array1.Concat(array2).ToArray();

Console.WriteLine("Array1: " + string.Join(",", array1));

Console.WriteLine("Array2: " + string.Join(",", array2));

Console.WriteLine("Concatenated Array: " + string.Join(",", concatenatedArray));
```

The resulting array contains the elements from array1 followed by the elements from array2. The string.Join() method is used to display the contents of the arrays in a read able format.