**Topic 5: Console App**

Top of Form

Bottom of Form

**Content**

**Parallel Arrays**

[Parallel Arrays](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286840_1)

Before you saw variables alone. To save several names, you needed to do the following:

// Names

string nameOne;

string nameTwo;

// Scores

double scoreOne;

double scoreTwo;

With arrays, you can declare the names in one array and the scores in another:

// Names

string[] names;

// Scores

double[] scores;

Before you could assign the score for nameOne in scoreOne and the score for nameTwo in scoreTwo. Thus allowing:

Console.Writeln(nameOne + " received a score of " + scoreOne);

Using parallel arrays, the index will line up the score and the name:

// Names & Scores

string[] names = {"Ayah", "Ananda"};

double[] scores = { 96.5, 99.7 };

// Print corresponding Names with Scores

Console.Writeln(names[0] + " received a score of " + scores[0]);

Console.Writeln(names[1] + " received a score of " + scores[1]);

How would you print multiple names for an array more efficiently?

**Using Array Class**

[Using Array Class](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286840_1)

In this module, you use arrays that you define and declare. They are a collection of items of the same data type. Any actions you use for your arrays, you are required to write the code.

Arrays are commonly used, as are the actions associated with arrays. There is an an Array class that extends the idea of the array and adds the commonly used actions as methods to create, manipulate, search and sort.

Array class is a data structure implemented in the C# library available to you by adding the compiler directive, which is by default included in your Console App.

using System;

As you will learn soon, a class has properties and methods. You need not fully understand that now to use the class and its methods.

Common methods:

[Search](https://docs.microsoft.com/en-us/dotnet/api/system.array.find?view=net-5.0)

// Declare array

int[] myIntArray = new int[5] { 10, 20, 30, 40, 50 };

// Search for an item, use lambda expression

int index = Array.FindIndex(myIntArray, element => element == 30);

Console.WriteLine("The items is at index " + index.ToString());

Other search methods: Exists(), Find(), FindAll(), FindIndex(), FindLast(), and FindLastIndex()

[Sort](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1.sort?view=netcore-3.1)

// Declare array

int[] myIntArray = new int[5] { 50, 20, 40, 10, 30 };

// Sort the array

Array.Sort(myIntArray);

Console.WriteLine("Sorted Array:");

foreach (var num in myIntArray)

{

Console.Write(" {0}", num);

}

**Using List Class**

[Using List Class](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286840_1)

A List is a collection of items of the same type that you can add, remove, search and sort. The [List class](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=netcore-3.1) is a data structure implemented in the C# library available to you by adding the compiler directive

using System.Collections.Generic;

As you will learn soon, a class has properties and methods. You need not fully understand that now to use the class and its methods.

Common methods:

[Search](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1.binarysearch?view=netcore-3.1)

List<string> nameList = new List<string>();

// Add names

nameList.Add("Ayah");

nameList.Add("Ananda");

// Find index of name using search method

int index = nameList.BinarySearch("Ayah");

// Print if found

if (index < 0)

{

Console.Writeln("Not found!");

}

else

{

Console.Writeln("Found at index: " + index.ToString());

}

[Sort](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1.sort?view=netcore-3.1)

List<string> nameList = new List<string>();

// Add names

nameList.Add("Ayah");

nameList.Add("Ananda");

// Sort list using sort method

nameList.Sort();

// Print sorted list

Console.WriteLine("List in sorted order: ");

foreach (var name in nameList)

{

Console.Write(" {0}", name);

}

**Search & Sort Arrays & Lists Console App Practice**

[Search & Sort Arrays & Lists Console App Practice](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286840_1)

This is for practice, give it a try before looking at the solution.

Write a console app to search and sort Arrays and Lists. Name the file ListsArrays.cs.

* + Method SortList() that takes a List and sorts it ascending order
    - Contains one line, a call the List library method sort
  + Method SortArray() that takes an array of double and sorts it descending order
    - You can use the Array Class method
  + Method SearchList()
    - Contains one line, a call the List library method search
    - Returns a boolean representing found
  + Method SearchArray()
    - You can use the Array Class search method
    - Returns boolean representing found
  + Main()
    - Declare and initialize a List
      * List containing subjects, named subjects
      * Add each in order: "Computer Science", "Web Development", "Cybersecurity" and "Networking"
    - Sort the list subjects alphabetically by calling sortList().

Prompt the user to input their major

* + - * Use SearchList() to find major
      * Display that major exists or does not exist
    - Declare and initialize parallel arrays
      * Array string to store student names, name the array studentName
      * Array of int to store studentId, name the array studentId
      * Initialize the above arrays to have {(Ayah, 322), (Manual, 323), (Mohamed, 334), (Vasavi, 325), (Rosa, 318)}
      * Array of int to store the index of their major, name the array majorSubject
      * Initialize the above array to have the following { 0, 2, 1, 0, 3 }
      * Array double to store scores, name it studentScore
    - Get User input
      * Display each student name, id, and major
      * Prompt for the score
      * Store the score in studentScore at appropriate index.
      * Don't forget input validation!
    - Prompt the user for a score to see if any student received that score
      * Use SearchArray() to see if it exists
    - Call sort array method for studentScore and print the scores
    - Print the list of scores

using System;

using System.Collections.Generic;

namespace Module6

{

class Program

{

public static void SortList(ref List<string> listToSort)

{

// Call List method to sort list

}

public static void SortArray(ref double[] arrayToSort)

{

// Call Array method to sort

}

public static void SearchList(List<string> list, string itemToSearch)

{

int existsInList;

existsInList = subjects.BinarySearch( // TODO

if (existsInList >= 0)

return true;

return false;

}

public static bool SearchArray(double[] studentScore, double itemToFind)

{

int existsInList;

existsInList = Array.FindIndex( //TODO

if (existsInList >= 0)

return true;

return false;

}

static void Main(string[] args)

{

// Declare and Initialize a list names subjects

// Add using List method: "Computer Science", "Web Development", "Cybersecurity" and "Networking"

// Sort the list by calling the method SortList()

// Get user input by prompting user to input a major

// Call SearchList() print a message to user if major exists

// Declare an array of string to store student names, name the array studentName

// Declare an array of int to store student ids, name the array studentId

// Initialize the above arrays to have {(Ayah, 322), (Manual, 323), (Mohamed, 334), (Vasavi, 325), (Rosa, 318)}

// Declare an array of int to store the index of their major, name the array majorSubject

// Initialize the above array to have the following { 0, 2, 1, 0, 3 }

// Declare an array of double to store scores, name it studentScore, add some valid test scores

// Print the name, id and major of each student

// Prompt user for input until a valid score to search is entered

// Call SearchArry and print if score is found

// Call SortArray method for any appropriate array and print the sorted array.

}

}

}

Possible Solution: [Module6ConsoleAppSolution.cs](https://dmacc.blackboard.com/bbcswebdav/pid-7286859-dt-content-rid-101466161_1/xid-101466161_1)

[**List and Array Method Quiz**](https://dmacc.blackboard.com/webapps/blackboard/content/launchAssessment.jsp?course_id=_102593_1&content_id=_7286863_1&mode=view)

This quiz covers comparing Lists and Arrays. It is not timed, you have 2 attempts.