**Topic 4: Lists**

Top of Form

Bottom of Form

**Content**

**Reading**

[Reading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286839_1)

Lists

* + Microsoft:docs.microsoft.com
    - [Collections (C#)](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/collections)
  + GeeksForGeeks: geeksforgeeks.com
    - [C# | List Class](https://www.geeksforgeeks.org/c-sharp-list-class/)
  + TutorialsPoint: tutorialspoint.com
    - [C# - List<T>](https://www.tutorialsteacher.com/csharp/csharp-list)

foreach

* + Microsoft:docs.microsoft.com
    - [foreach](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/using-foreach-with-arrays)
  + GeeksForGeeks: geeksforgeeks.com
    - [C# using foreach loop in arrays](https://www.geeksforgeeks.org/c-sharp-using-foreach-loop-in-arrays/)
  + TutorialsPoint: tutorialspoint.com
    - [foreach loop in C#](https://www.tutorialspoint.com/foreach-loop-in-chash)
  + W3Schools: w3schools.com
    - [C# foreach](https://www.w3schools.com/cs/cs_for_loop.asp)

NOTES

* + [Lists using Generics](https://dmacc.blackboard.com/bbcswebdav/pid-7286846-dt-content-rid-101466162_1/xid-101466162_1) [Lists using Generics - Alternative Formats](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286839_1)
  + [ArrayLists in C#](https://dmacc.blackboard.com/bbcswebdav/pid-7286846-dt-content-rid-101466163_1/xid-101466163_1) [ArrayLists in C# - Alternative Formats](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286839_1)

**Lists**

[Lists](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286839_1)

Lists are similar to arrays, but improves on a few of the limitations of arrays.

* + List does not need a size to be initialized
  + List can re-size automatically (grow with addition, shrink with removal)

List is collection and there are several for you to use. These collections are parts of class libraries available to you. Instead of always declaring arrays for storage and rewriting searching, sorting and other algorithms, you can use a collection. These collections are classes, so they have methods (search, find, sort, etc) depending on the collection. One other example of a collection available is Array (notice this capitalized). Array is similar to the arrays you have declared, with added methods. Do some searching on your own to find other C# collections.

**List search**

List<string> source = new List <string> { "swim", "bike", "run" };

string exercise = "Yoga"; int position = source.IndexOf(exercise); //will return -1, not found

position = source.IndexOf("bike"); //will return 1

**List copy**

List<string> source = new List<string> { "swim", "bike", "run" };

List<string> target; // no size needed

// use the constructor to copy

target = new List<string>(source);

**foreach Loop**

[foreach Loop](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286839_1)

Recall the foreach loop you learned for accessing arrays.

**Array**

string[] names = new string[] = { "Ayah", "Wyatt", "Natalie", "Nilah" };

foreach (string name in names) {

Console.WriteLine(name);

}

int[] ages = new ages[] = { 21, 46, 18, 35 };

foreach (int age in ages) {

Console.WriteLine(age);

}

You can use this same loop to access a collection such as a List.

This loop structure allows the programmer to access each element in an array or list without using the index:

// Collection: collection of items, arrays, Lists, etc

foreach (type variable in Collection) {

// code here

}

**List**

List<string> names = new List<string>( ) { "Morgan", "Wyatt", "Natalie", "Nilah" };

foreach (string name in names) {

Console.WriteLine(name);

}

[**Sorting Algorithms**](https://brilliant.org/wiki/sorting-algorithms/)

[Sorting Algorithms](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286839_1)

Check out some sorting algorithms!