Lesson 33 - Collections III

A **collection** is used to group related objects. Unlike an array, it is **dynamic** and can also group objects. A collection can grow and shrink to accommodate any number of objects. Collection classes are organized into **namespaces** and contain built in methods for processing elements within the collection.

A collection **organizes** related data in a computer so that it can be used efficiently. Different kinds of collections are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, `Dictionaries` are used to represent connections on social websites (such as Twitter, Facebook), `queues` can be used to create task schedulers, `HashSets` are used in searching algorithms, etc.

A collection typically includes methods to `add`, `remove`, and `count` objects. The `for` statement and the `foreach` statement are used to **iterate** through collections. Since a collection is a class you must first declare an instance of the class before you can add elements to that collection.

```
List<int> list = new();
```

> Collections provide a more flexible way to work with groups of objects. Unlike arrays, the group of objects you work with can grow and shrink dynamically as the needs of the application change.

Generic Collections

Generic collections are the preferred type to use as long as every element in the collection is of the same data type. Only desired data types can be added to a generic collection and this is enforced by using strong typing which reduces the possibility of errors.
The .NET Framework provides a number of generic collection classes, useful for storing and manipulating data.
The `System.Collections.Generic` namespace includes the following generic collections:

- `List<T>`
- `Dictionary<TKey, TValue>`
- `SortedList<TKey, TValue>`

- `Stack<T>`
- `Queue<T>`
- `Hashset<T>`

To access a generic collection in your code, you will need to include the statement:

```
using Systems.Collections.Generic;
```

Non-Generic Collections

Non-generic collections can store items that are of type Object. Since an Object data type can refer to any data type, you run the risk of unexpected outcomes. Non-generic collections may also be slower to access as well as execute.
The `System.Collections` namespace includes the following non-generic collections:

- `ArrayList`
- `SortedList`
- `Stack`
- `Queue`
- `Hashtable`
- `BitArray`

> Because non-generic collections are error prone and less performant, it is recommended to always use generic collections from the System.Collections.Generic namespace if available and to avoid using legacy collections from the System.Collections namespace.