Collections in C#

By

Narasimha Rao T

Microsoft.Net FSD Trainer

Professional Development Trainer

tnrao.trainer@gmail.com

Why Use Collections Over Arrays?

Arrays

- **Fixed size** once declared.

- Best when you know the exact number of elements.

- Offers fast access via index.

```
int[] numbers = new int[5];
numbers[0] = 10;
```

Limitations of Arrays:

- Size can't be changed at runtime.

- No built-in methods for dynamic operations like adding or removing.

- Not ideal for real-time or large-scale applications where size varies.

Array vs ArrayList vs List<T>

Feature	Array	ArrayList	List<T>
Type Safety	(strongly typed)	(object type)	(generic, strongly typed)
Performance	Fast	Slower (boxing/unboxing)	Fast (type-safe)
Flexibility	Fixed size	Dynamic	Dynamic
Usage Example	`int[] arr = new int[3];`	`ArrayList list = new ArrayList();`	`List<int> list = new List<int>();`

Introduction to Generic Collections

What Are Generics?

- Allow classes and methods to operate on **any data type** while maintaining **type safety**.

- Avoid boxing/unboxing.

- Enable code reusability.

```
List<string> names = new List<string>();
```

Benefits:

- Compile-time type checking.

- Better performance.

- Code readability.

List<T>

Description:

- Represents a strongly typed list of objects.

- Resizable array.

```
List<int> numbers = new List<int>();
numbers.Add(10);
numbers.Add(20);
```

Dictionary<TKey, TValue>

Description:

- Stores key-value pairs.

- Fast lookups by key.

```
Dictionary<string, int> ages = new Dictionary<string, int>();
ages["Alice"] = 30;
ages["Bob"] = 25;
```

Common Operations on Collections

Add

```
list.Add("New Item");
```

Remove

```
list.Remove("Item");
```

Contains

```
if (list.Contains("Item")) {
    // Do something
}
```

Iteration

```csharp
foreach (string item in list) {
    Console.WriteLine(item);
}
```

Summary

- Arrays are limited by fixed size.

- Collections offer flexibility, dynamic sizing, and richer features.

- Generics improve performance and safety.

- Choose the right collection (List, Dictionary, Queue, Stack) based on your use case.

Q & A

Narasimha Rao T

tnrao.trainer@gmail.com