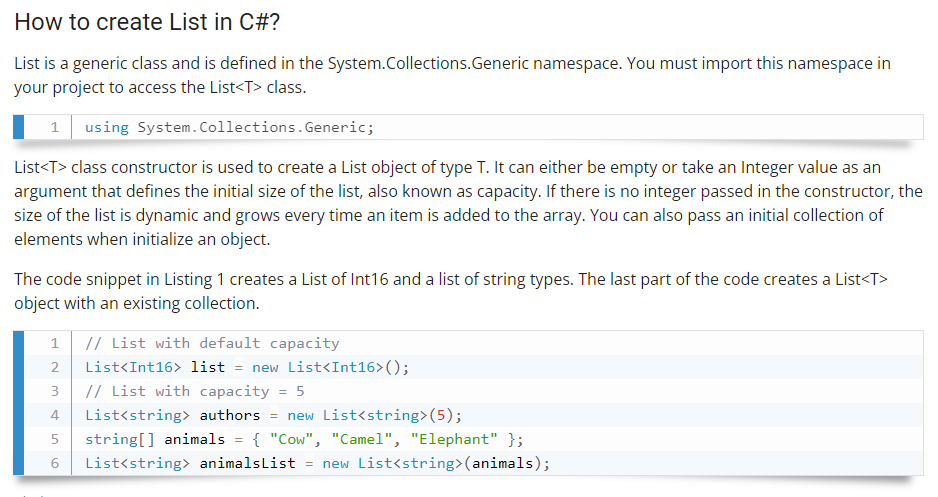
04 Generics  
Test your Knowledge  
1. Describe the problem generics address.

The following are some limitations of generics:

* Generic types can be derived from most base classes, such as [MarshalByRefObject](https://docs.microsoft.com/en-us/dotnet/api/system.marshalbyrefobject) (and constraints can be used to require that generic type parameters derive from base classes like [MarshalByRefObject](https://docs.microsoft.com/en-us/dotnet/api/system.marshalbyrefobject)). However, .NET does not support context-bound generic types. A generic type can be derived from [ContextBoundObject](https://docs.microsoft.com/en-us/dotnet/api/system.contextboundobject), but trying to create an instance of that type causes a [TypeLoadException](https://docs.microsoft.com/en-us/dotnet/api/system.typeloadexception).
* Enumerations cannot have generic type parameters. An enumeration can be generic only incidentally (for example, because it is nested in a generic type that is defined using Visual Basic, C#, or C++). For more information, see "Enumerations" in [Common Type System](https://docs.microsoft.com/en-us/dotnet/standard/base-types/common-type-system).
* Lightweight dynamic methods cannot be generic.
* In Visual Basic, C#, and C++, a nested type that is enclosed in a generic type cannot be instantiated unless types have been assigned to the type parameters of all enclosing types. Another way of saying this is that in reflection, a nested type that is defined using these languages includes the type parameters of all its enclosing types. This allows the type parameters of enclosing types to be used in the member definitions of a nested type. For more information, see "Nested Types" in [MakeGenericType](https://docs.microsoft.com/en-us/dotnet/api/system.type.makegenerictype).

2. How would you create a list of strings, using the generic List class?

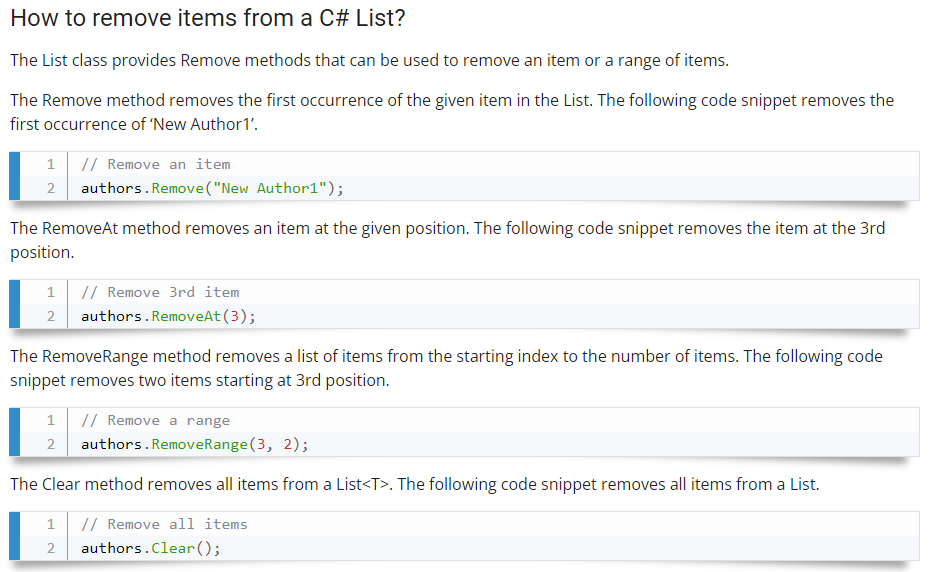
  
3. How many generic type parameters does the Dictionary class have?

* the [System.Collections.Generic.Dictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2) class can contain two types: keys and values. Because a generic type definition is only a template, you cannot create instances of a class, structure, or interface that is a generic type definition.
* Generic type parameters, or type parameters, are the placeholders in a generic type or method definition. The [System.Collections.Generic.Dictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2) generic type has two type parameters, TKey and TValue, that represent the types of its keys and values.

4. True/False. When a generic class has multiple type parameters, they must all match.  
5. What method is used to add items to a List object?



  
6. Name two methods that cause items to be removed from a List.

  
7. How do you indicate that a class has a generic type parameter?

A generic type is declared by specifying a type parameter in an angle brackets after a type name, e.g. TypeName<T> where T is a type parameter.  
8. True/False. Generic classes can only have one generic type parameter.  
9. True/False. Generic type constraints limit what can be used for the generic type.  
10. True/False. Constraints let you use the methods of the thing you are constraining to.

Practice working with Generics

1. Create a custom Stack class MyStack<T> that can be used with any data type which  
has following methods  
1. int Count()  
2. T Pop()  
3. Void Push()

2. Create a Generic List data structure MyList<T> that can store any data type.  
Implement the following methods.  
1. void Add (T element)  
2. T Remove (int index)  
3. bool Contains (T element)  
4. void Clear ()  
5. void InsertAt (T element, int index)  
6. void DeleteAt (int index)  
7. T Find (int index)

3. Implement a GenericRepository<T> class that implements IRepository<T> interface  
that will have common /CRUD/ operations so that it can work with any data source  
such as SQL Server, Oracle, In-Memory Data etc. Make sure you have a type constraint  
on T were it should be of reference type and can be of type Entity which has one  
property called Id. IRepository<T> should have following methods  
1. void Add(T item)  
2. void Remove(T item)

3. Void Save()  
4. IEnumerable<T> GetAll()  
5. T GetById(int id)