# C# List<T> Methods Example

This document provides a comprehensive C# program demonstrating the usage of various List<T> methods with proper comments.

using System;  
using System.Collections.Generic;  
using System.Linq;  
  
class Program  
{  
 static void Main()  
 {  
 // Initialize a List of integers  
 List<int> numbers = new List<int> { 5, 3, 8, 1, 9 };  
  
 // 1. Add an element to the list  
 numbers.Add(10); // Adds 10 at the end  
 Console.WriteLine("After Add(10): " + string.Join(", ", numbers));  
  
 // 2. Add multiple elements using AddRange()  
 numbers.AddRange(new List<int> { 12, 15, 18 });  
 Console.WriteLine("After AddRange(): " + string.Join(", ", numbers));  
  
 // 3. Insert an element at a specific index  
 numbers.Insert(2, 7); // Inserts 7 at index 2  
 Console.WriteLine("After Insert(2, 7): " + string.Join(", ", numbers));  
  
 // 4. Remove an element by value  
 numbers.Remove(3); // Removes first occurrence of 3  
 Console.WriteLine("After Remove(3): " + string.Join(", ", numbers));  
  
 // 5. Remove an element by index  
 numbers.RemoveAt(4); // Removes element at index 4  
 Console.WriteLine("After RemoveAt(4): " + string.Join(", ", numbers));  
  
 // 6. Remove a range of elements (from index 2, remove 2 elements)  
 numbers.RemoveRange(2, 2);  
 Console.WriteLine("After RemoveRange(2,2): " + string.Join(", ", numbers));  
  
 // 7. Check if a value exists in the list  
 Console.WriteLine("Contains(8): " + numbers.Contains(8)); // true  
 Console.WriteLine("Contains(100): " + numbers.Contains(100)); // false  
  
 // 8. Find the index of a specific element  
 int index = numbers.IndexOf(9);  
 Console.WriteLine("Index of 9: " + index);  
  
 // 9. Sort the list in ascending order  
 numbers.Sort();  
 Console.WriteLine("After Sort(): " + string.Join(", ", numbers));  
  
 // 10. Sort in descending order using custom comparison  
 numbers.Sort((a, b) => b.CompareTo(a));  
 Console.WriteLine("After Descending Sort(): " + string.Join(", ", numbers));  
  
 // 11. Reverse the list  
 numbers.Reverse();  
 Console.WriteLine("After Reverse(): " + string.Join(", ", numbers));  
  
 // 12. Find the first element greater than 4  
 int found = numbers.Find(x => x > 4);  
 Console.WriteLine("Find(x > 4): " + found);  
  
 // 13. Find all even numbers  
 List<int> evenNumbers = numbers.FindAll(x => x % 2 == 0);  
 Console.WriteLine("FindAll(even numbers): " + string.Join(", ", evenNumbers));  
  
 // 14. Iterate over list using ForEach()  
 Console.WriteLine("Using ForEach():");  
 numbers.ForEach(num => Console.Write(num + " "));  
 Console.WriteLine();  
  
 // 15. Convert list to array  
 int[] numArray = numbers.ToArray();  
 Console.WriteLine("Converted to Array: " + string.Join(", ", numArray));  
  
 // 16. Clear all elements from the list  
 numbers.Clear();  
 Console.WriteLine("After Clear(): Count = " + numbers.Count);  
 }  
}