Accessing Out of Bounds

Effectively, we can add an infinite number of items to a list:

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

for (int i = 0 ; i < 1000; i++)
{
  numbers.Add(i);
}
```

We can access and edit them using bracket notation:

```
int numberFive = numbers[4];
numbers[4] = 523;
```

However this doesn’t mean that we can access any random index. If we try to access the 1001st element in this list, we’ll get an error:

```
Console.WriteLine(numbers[1001]);
```

The above code causes the error:

```
Unhandled Exception:
System.ArgumentOutOfRangeException: Index was out of range. Must be non-
negative and less than the size of the collection.
```

We can only access indices which have been added to the list. The last available index in the `numbers` list would be `999`.

Here are two tips to avoid this issue:

> Imagine the list growing every time we add a number and shrinking every time we remove a number. Unlike arrays, there is no set length.
>
> Check the `Count` of your lists before accessing an index, as shown below.

```
int index = 1001;
if (index < numbers.Count)
{
  Console.WriteLine(numbers[index])
}
```

☑Instructions

1.

This code has two issues. Fix each erroneous statement by changing the index it uses.

Hint

The smallest possible index in `marathons` is `0` and the largest possible is `3`.