# Working with Collection Lesson

## Using Collection built-in methods and Linq commands

Step 1.

Have the students create a console program in the OOPS review solution.

Step 2.

Add a reference to the class library in your solution.

Step 3.

Write a method to create a collection of a class. This collection is one for my Employment class. Here you can point out that Emloyment is unknown, even with the reference. Use the intelli-sense light bulb to select the using statement.

List<Employment> CreateCollection()

{

List<Employment> newCollection = new List<Employment>();

newCollection.Add(new Employment("PG I", SupervisoryLevel.Entry,

DateTime.Parse("May 1, 2010"), 0.5));

newCollection.Add(new Employment("PG II", SupervisoryLevel.TeamMember,

DateTime.Parse("Nov 1, 2010"), 3.2));

newCollection.Add(new Employment("PG III", SupervisoryLevel.TeamLeader,

DateTime.Parse("Jan 6, 2014"), 8.6));

newCollection.Add(new Employment("SP I", SupervisoryLevel.Supervisor,

DateTime.Parse("Jul 22, 2022"), 1.8));

return newCollection;

}

Step 4.

Add code to call the method and display the data in the collection.

List<Employment> employments = new List<Employment>();

employments = CreateCollection();

foreach (var item in employments)

{

Console.WriteLine($"{item.ToString()}");

}

Step 5.

Use a method they should already be familiar with for collections, .Count(). Discuss why collections have methods and properties. Created for developers so they don’t have to do it themselves.

Console.WriteLine($"Number of item in the collections is {employments.Count()}");

Step 6.

Use a loop to find an instance in the collection. Discuss logic which they should know.

Employment employment = null;

foreach (var item in employments)

{

if (item.Title.Equals("PG II"))

employment = item;

}

if (employment != null)

Console.WriteLine($"\nA PG II instance found {employment.ToString()} via a loop\n");

Step 6.

Now start showing the “magic”. Use collection method .Find(predicate).

“What if I could do the same process but in one line of code. Anyone interested????

employment = null;

employment = employments.Find(p => p.Title.Equals("PG II"));

if (employment != null)

Console.WriteLine($"A PG II instance found {employment.ToString()}via Find(…)");

Discuss and breakdown the predicate. I hold my cursor on the Find and discuss the tool tip displayed.

Find returns an instance or null.

employment = null;

employment = employments.Find(p => p.Title.Equals("PG"));

if (employment != null)

Console.WriteLine($"A PG instance found {employment.ToString()}");

else

Console.WriteLine($"No PG instance found");

Step 7.

Play with .Contains() instead of .Equals(). Discuss the difference between the two. Point out which instance is returned.

employment = null;

employment = employments.Find(p => p.Title.Contains("PG"));

if (employment != null)

Console.WriteLine($"\nA PG instance found {employment.ToString()} via Contains(...)\n");

else

Console.WriteLine($"\nNo PG instance found\n");

Step 8.

Introduce .Where(predicate). Discuss that Where expects to return a collection, not a single instance like Find(). Where(…) is a Linq method.

employment = null;

employment = employments.Where(p => p.Title.Contains("PG"));

if (employment != null)

Console.WriteLine($"\nA PG instance found {employment.ToString()} via Contains(...)\n");

else

Console.WriteLine($"\nNo PG instance found\n");

Need to alter the code. Discuss that the default return datatype is IEnumerable<T>. To convert to a List<T> one needs to add .ToList().

List<Employment> searchEmployments = null;

searchEmployments = employments.Where(p => p.Title.Contains("PG")).ToList();

if (searchEmployments.Count() > 0)

Console.WriteLine($"\nA PG instance found. {searchEmployments.Count()} instances found.\n");

else

Console.WriteLine($"\nNo PG instances found\n");

You could do additional Where examples.

Step 9.

Is there a method that searches and returns a Boolean. Discuss .Any(predicate). Returns a boolean value and NOT a physical collection. Any(…) is a Linq method. Great if you just need know if something exists or not and NOT actually get a collection.

if (employments.Any(p => p.Title.Contains("PG")))

Console.WriteLine($"\nA PG instance found using Any(...). \n");

else

Console.WriteLine($"\nNo PG instances found\n");

if (employments.Any(p => p.Title.Contains("PG IV")))

Console.WriteLine($"\nA PG IV instance found using Any(...). \n");

else

Console.WriteLine($"\nNo PG IV instances found\n");