# Determine the Type of Data Contained within Collections



Paul D. Sheriff
Business / IT Consultant

psheriff@pdsa.com | www.pdsa.com



### **Module Goals**



### **Answer questions about a collection...**

- Do All() items meet a condition
- Do **Any**() items meet a condition
- Collection Contains() an item?

**Use an EqualityComparer<T> for classes** 



# All() Method



# **Uses of All() Method**

Are all products' price greater than their cost?

Do all sales orders have a quantity greater than or equal to 1?

Do all customers have a zero balance?



```
IEnumerable<T>.All(predicate);

products.All(prod => prod.ListPrice > prod.StandardCost);
```

- ◄ All() searches the entire collection
- **◆** Determines if all items match the condition
- Do all products' list price exceed their cost?



All() method

# Any() Method



# **Uses of Any() Method**

Do any sales orders have a quantity greater than 10

Do any sales orders have a total greater than 10k?

Do any customers have a credit balance?



```
IEnumerable<T>.Any(predicate);
sales.Any(sale =>
    sale.LineTotal > 10000);
```

- Any() method searches entire collection
- Determines if any items in collection match the condition
- Do any sales have a line total greater than 10,000?



Any() method

# **Contains() Method**



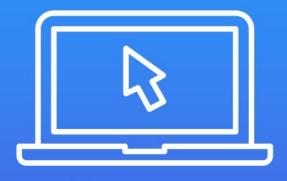
## Contains() Method

Searches collection to see if a value exists

For simple data type collections such as int, decimal, string, etc.

Checks if value in the collection is equal to value you are searching for





### **Contains using an integer list**

## **Contains() with Objects**

Default is to compare object references

You probably want to look at the value in one or more properties of an object

Need to create EqualityComparer<T > class



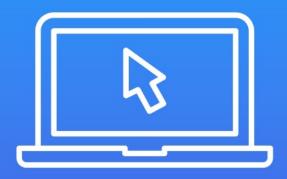
```
public class ProductIdComparer :
             EqualityComparer<Product>
  public override bool Equals(Product x,
                              Product y)
    return (x.ProductID == y.ProductID);
  public override int
    GetHashCode(Product obj) {
    return obj.ProductID.GetHashCode();
```

■ Inherit from EqualityComparer<Product>

- Override Equals(product 1, product 2) method
- Return true if both match

- Override GetHashCode() method





Contains() using a comparer class

# Module Summary



All() checks if all items match a predicate

Any() checks if any items match a predicate

Contains() with a comparer make it easy to search object property values

Need to build a comparer class for each type of search you wish to perform

**Up Next:** 

# Determine Differences Between Two Collections

