**Assignment 02 LINQ**

Note: Use ListGenerators.cs & Customers.xml

# LINQ - Set Operators

1. Find the unique Category names from Product List

2. Produce a Sequence containing the unique first letter from both product and customer names.

3. Create one sequence that contains the common first letter from both product and customer names.

4. Create one sequence that contains the first letters of product names that are not also first letters of customer names.

5. Create one sequence that contains the last Three Characters in each name of all customers and products, including any duplicates

# LINQ - Aggregate Operators

1. Get the total units in stock for each product category.

2. Get the cheapest price among each category's products

3. Get the products with the cheapest price in each category (Use **Let**)

4. Get the most expensive price among each category's products.

5. Get the products with the most expensive price in each category.

6. Get the average price of each category's products.

# LINQ - Partitioning Operators

1. Get the first 3 orders from customers in Washington

2. Get all but the first 2 orders from customers in Washington.

3. Return elements starting from the beginning of the array until a number is hit that is less than its position in the array.

int [] numbers = {5, 4, 1, 3, 9, 8, 6, 7, 2, 0};

4.Get the elements of the array starting from the first element divisible by 3.

int [] numbers = {5, 4, 1, 3, 9, 8, 6, 7, 2, 0};

5. Get the elements of the array starting from the first element less than its position.

int [] numbers = {5, 4, 1, 3, 9, 8, 6, 7, 2, 0};

# LINQ - Quantifiers

1. Determine if any of the words in dictionary\_english.txt (Read dictionary\_english.txt into Array of String First) contain the substring 'ei'.

2. Return a grouped a list of products only for categories that have at least one product that is out of stock.

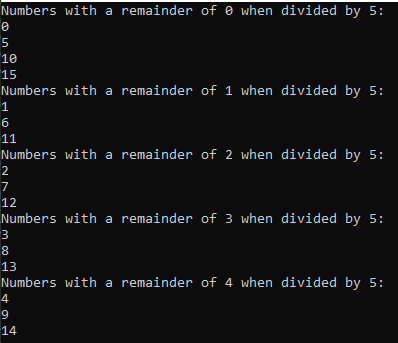
3. Return a grouped a list of products only for categories that have all of their products in stock.

# LINQ – Grouping Operators

1. Use group by to partition a list of numbers by their remainder when divided by 5

**List<int> numbers = new list<int> {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};**

**Output:**



1. Uses group by to partition a list of words by their first letter.

Use dictionary\_english.txt for Input

1. Consider this Array as an Input

**String [] Arr = {"from", "salt", "earn", " last", "near", "form"};**

Use Group By with a custom comparerthatmatches words that are consists of the same Characters Together

**Output:**

