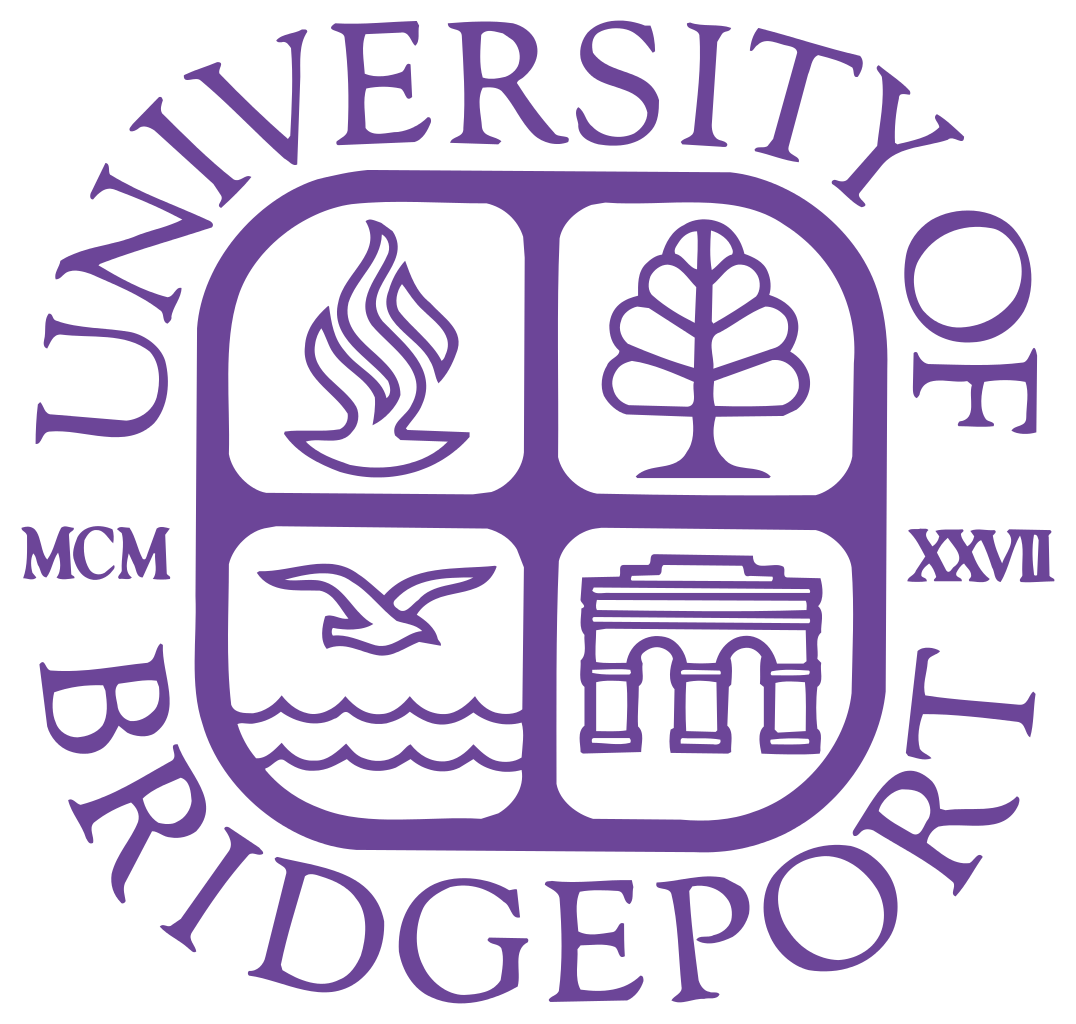
******

**Bashar Alhafni**

**Assignment #5**

**Objective:**

a) Suppose there is a Product class with the following fields:

ProductId : int

ProductName : string

Price : decimal

Category : an enum which can have one of the three possibilities of

SPORTS, BOOKS, ELECTRONICS

Description : string

An example of a Product object will be

{12341, “Calculator”, 24.95m, ELECTRONICS, “Solar powered Scientific”}

Suppose there is an OnlineStore class that has an array of Products. Initialize 3 products

belonging to each category of SPORTS, BOOKS and ELECTRONICS (total of 9 products in the

constructor for OnlineStore). Then develop the following indexers in a Test Driven Development

(TDD) style.

1. Search by ProductId

2. Search by ProductName

3. Search by partial description e.g., if you searched by “solar”, the calculator will match.

4. Search by category.

Note that the search by partial description and search by category will return an array of

products.

Show the unit tests for all above cases.

**Approach Explanation:**

I created indexers for every specified field that we need to search by

1) I created an indexer that will take the id and return the product that has this specific ID

2) I created an indexer that will take a name as a string and return the product with this specified name

3) I created an indexer that will take the partial description as a string and an enumeration called SearchType just to overcome the overloading issue of having 2 methods with the same number of parameters and the same types. This indexer returns a list of products which match the given partial description

4) I created an indexer that will take Category, which is an enumeration and return all the products which belong to this category.

I also created a test project called SearchTest where I test all the indexers which were written in the OnlineStore class. I followed the TDD approach in order to develop my solution

**Screen shot:**

