Assignment2

1. Create a delegate called **ProductFilter** that takes a **Product** as input and returns a **bool**.
2. Implement an extension method called **Filter** for **IEnumerable<Product>** that uses the **ProductFilter** delegate to filter products.
3. Implement an extension method called **Transform** for **IEnumerable<Product>** that takes a **delegate** to transform a Product into a **string** (for displaying product details).
4. Create methods that return **ProductFilter** delegates for the following conditions:
    - Products with a price greater than a specified value
    - Products in a specific category
    - Products whose name contains a specific string (case-insensitive)
5. Use the **Filter** and **Transform** extension methods to:
    - Display all products with a price greater than $500
    - Display all products in the "Furniture" category
    - Display all products whose name contains "a"

Hints:

- For the **Filter** method, use the **yield** keyword to create an iterator.
- For the **Transform** method, consider using a delegate that takes a Product and returns a string.
- Use the following Class Product and List

```
    List<Product> products = new List<Product>  {
        new Product { Name = "Laptop", Price = 1200, Category = "Electronics" },
        new Product { Name = "Desk Chair", Price = 250, Category = "Furniture" },
        new Product { Name = "Coffee Maker", Price = 100, Category = "Appliances" },
        new Product { Name = "Bookshelf", Price = 180, Category = "Furniture" },
        new Product { Name = "Smartphone", Price = 800, Category = "Electronics" }
    };
public class Product {
    public string Name { get; set; }
    public decimal Price { get; set; }
    public string Category { get; set; }
}
```

By Ali A. Ahmed