

Select with Anonymous Types: prints the name of every product in the product list along with the category of the product and its unit price

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class MainClass {
public static void Main() {

List products = GetProductList();

var productInfos =
from p in products
select new { p.ProductName, p.Category, Price = p.UnitPrice };

Console.WriteLine("Product Info:");
foreach (var productInfo in productInfos) {
Console.WriteLine("{0} is in the category {1} and costs {2}
per unit.", productInfo.ProductName, productInfo.Category,
productInfo.Price);
}
}

static List GetProductList() {
List empTree = new List();
empTree.Add(new Product { ProductName = "A", Category = "0",
UnitPrice = 12, UnitsInStock = 5, Total = 36, OrderDate = new
```

```

DateTime(2005, 1, 1), Id = 1 }));
empTree.Add(new Product { ProductName = "B", Category = "0",
UnitPrice = 2, UnitsInStock = 4, Total = 35, OrderDate = new
DateTime(2005, 1, 1), Id = 1 }));
empTree.Add(new Product { ProductName = "C", Category = "0",
UnitPrice = 112, UnitsInStock = 3, Total = 34, OrderDate = new
DateTime(2005, 1, 1), Id = 1 }));
empTree.Add(new Product { ProductName = "D", Category = "0",
UnitPrice = 112, UnitsInStock = 0, Total = 33, OrderDate = new
DateTime(2005, 1, 1), Id = 1 }));
empTree.Add(new Product { ProductName = "E", Category = "0",
UnitPrice = 1112, UnitsInStock = 2, Total = 32, OrderDate =
new DateTime(2005, 1, 1), Id = 1 }));
empTree.Add(new Product { ProductName = "F", Category = "0",
UnitPrice = 11112, UnitsInStock = 0, Total = 31, OrderDate =
new DateTime(2005, 1, 1), Id = 1 }));
return empTree;
}
}
class Product : IComparable {
public string ProductName { get; set; }
public string Category { get; set; }
public int UnitPrice { get; set; }
public int UnitsInStock { get; set; }
public int Total { get; set; }
public DateTime OrderDate { get; set; }
public int Id { get; set; }

public override string ToString() {
return String.Format("Id: {0}, Name: {1} , Category: {3}",
this.Id, this.ProductName, this.Category);
}
int IComparable.CompareTo(Product other) {
if (other == null)
return 1;
if (this.Id > other.Id)
return 1;
}

```

```
if (this.Id < other.Id) return -1; return 0; } } [/csharp]
```