Programming the .Net Framework using C#

# Formatting Ex

Object Oriented programming in C#

## Instructions

Build a **Console Application** that shows a user a list of products, asks the user to chose a product, and once a product was selected, prints the product invoice.

Ask the user for his or her name.

Your program should display on the Console window at least three products, each with:

* Product name (string)
* Product price (double)
* Product warranty expiration date (DateTime object) (תוקף סוף אחריות)

The user should select which of the products to buy.

Once the user selected a product, create string that represents the invoice (חשבונית) and display it on the Console Window

## Detailed Instructions

1. Define a Product class.
   1. Each product has three fields: (make sure fields are private)

* Product name (string)
* Product price (double)
* Product warranty expiration date (DateTime)
  1. Product class should have a public constructor that receives (string name, double price, and int numDays). This constructor will set the product name to name, the product price to price (should be a positive number, otherwise the default is 100),

and the warrantyExpirationDate to purchase date + the numDays.

The DateTime requirenebt can be done by using DateTime property Now – returns the current date, and AddDays(int) method which returns the date + added number of days.

See DateTime documentation for details.

Example: if \_warrantyDate is of type DateTime and numDays is of type int:

\_warrantyDate = DateTime.Now;// initialize the date as today

\_warrantyDate =\_warrantyDate.AddDays ( numDays);

// adds to the date the number of days,

// so now its today + number of days

* 1. Override ToString() for product class to show the name, price, warranty date
  2. Add three public Get properties or methods (whichever you know better) to get the product name (string), price (double) and warrantyDate (DateTime)

1. It is advised to create a Program class, even if you use .Net 6. (instead of what the Console app gives in Program.cs file) The class should look as follows: (what should you do in order to use the Product class from main? Namespace?)

class Program

{

// here you can add more functions

// Main is the entry point to the program

static void Main(string[] args)

{

// the following line make sure you work in US culture

// even if your computer is set to Israel Hebrew

// add it as first line in Main

Thread.CurrentThread.CurrentCulture = new

System.Globalization.CultureInfo("en-US");

// enter your main code here.

}

}

1. In your Program create at least three such products
2. Ask the user for his or her name using Console WriteLine and ReadLine methods
3. Show the user the products you created on the console window, (using Console.WriteLine and the “ToString()” method of the produces you created ) , and ask him/her to choose one. Make sure their choice is legal . (for example: if you gave 3 choices, and they give you a number, make sure its between 1-3)
4. Once a product was chosen, create a string representing the invoice**. Use string formatting we learned in class**. The invoice should show the following string: (each item in a separate line, use \n)

* Customer name
* Product name
* Product price**, shown in currency format**
* Product warranty expiration date **, in Full date/time pattern (long time).**
* **Note the last two requirements for special formatting**. Consult DateTime Formatting options in the course PDF (end of chapter 2) or in the DateTime documentation online.

1. Display the invoice **on the Console window**

**Note:** if your computer is set to Hebrew culture, the console window representation of currency and date might show ????????? – In order to correct this, add the following line to your Main function (see above in demonstrating the Program class structure)

Thread.CurrentThread.CurrentCulture = new

System.Globalization.CultureInfo("en-US");

## Features in C# and in the course that may be used in this project: (from modules 1 and 2 of the course)

* String formatting
* Console class methods
* Reading documentation (for example DateTime)
* Using namespaces
* Building a very simple class (based only on previous courses) .
* Overriding ToString() object method.

## Important Notes

* Make sure you use string formatting as we learned in class
* Make sure your code is readable, using meaningful names, indentation and documentation.
* Avoid code duplication
* Use methods to break your code to small functional units.