COMP123 – Programming 2

Assignment 5

Dollar Computers

Due Week #14 (Friday August 16, 2019) @midnight.

Value 10%

Dollar Computers **Maximum Mark:** 77

Overview: Create a multi-form project that simulates a computer purchase. The program initially gives the user an option to "start a new order" or "load a saved order". If the user selects "start a new order" the program allows the user to select computer hardware from a dynamic grid generated from a database connection. The user will then select computer hardware from a form control. The selected hardware components will display on a separate form and the user will have the option to save the configuration to a file. If the user selects "load a saved order" the program displays a dialog box that allows the user to load a previously saved hardware configuration. The application will then calculate the cost of the Computer (including taxes) in a separate form.

ProductInfoForm Menu Diagram

File	Edit
Open	Select Another Product
Save	
Exit	

OrderForm Menu Diagram

File	Edit	Help
Print	Back	About
Exit		

Instructions:

The program should contain 4 forms: **StartForm** (where the user selects between starting a new order or loading a saved order), **SelectForm** (where the user selects computer hardware), **ProductInfoForm** (where the computer hardware components are displayed and where the user has the option to load or save the component configuration), **OrderForm** (where the user is shown the cost of the hardware and can print his order).

(27 Marks: GUI, 38 Marks: Functionality, 3 Marks: Solution Structure, 5 Marks: Internal Documentation, 4 Marks: Version Control)

1. **StartForm: (SubTotal: 8 Marks: GUI, 6 Marks: Functionality)**
    a. Create a **Splash Screen** for the project that includes a company logo for "Dollar Computers". Ensure that the form is displayed for a minimum of 3 seconds. (1 Mark: GUI, 1 Mark: Functionality)
    b. Add a **button control** that gives the user an option to **start a new order**. The button will hide **StartForm** and show **SelectForm** (1 Mark: GUI, 1 Mark: Functionality).
    c. Add a **button control** that gives the user an option to **load a saved order**. The button will hide **StartForm**, show **ProductInfoForm** and display an **OpenFileDialog** box that allows the user to load a previously saved file. The **OpenFileDialog** Method will be contained in the **ProductInfoForm** (1 Mark: GUI, 2 Marks: Functionality).
    d. Add a **button control** that gives the user an option to **Exit the program**. The button will **terminate the application** (1 Mark: GUI, 1 Mark: Functionality).
    e. Add a **PictureBox control** to the form that displays an interesting image relating to "Dollar Computers" from the project's resource library (1 Mark: GUI).
    f. Define a new **default icon** for the project (Project Properties -> Application Tab). Ensure the icon is no larger than 32px x 32px. The icon needs to have a .ico extension (hint: go to http://converticon.com to get this done) (1 Mark: GUI, 1 Mark: Functionality).
    g. Set the **ControlBox** properties of the form to False (1 Mark: GUI).
    h. Set the **StartPosition** properties of the form and the **SplashScreen** to **CenterScreen** (1 Mark: GUI).
2. **SelectForm: (SubTotal: 6 Marks: GUI, 11 Marks: Functionality)**
    a. Add a **label** to the top of the form that lets the user know he's about to make a computer hardware selection (1 Mark: GUI).
    b. Either locally (localdb) or on MS Azure create a Database named **DollarComputers**. Use the SQL Database file provided to generate the **Products** table for your database (2 Marks: Functionality).
    c. Use the **Entity Framework** to connect to the Database Server (either localdb or on MS Azure). Ensure you use the **Code First from Database** when creating your Entity data model and the Products Model (2 Marks: Functionality).
    d. Create a **DataGridView** control. Ensure the **DataGridView's** ReadOnly property is set to True. (1 Mark: GUI).
    e. Use the Products Table as a **DataSource** for the DataGridView. (1 Mark: Functionality).
    f. Add a **label** and **a text box** to the bottom of the form that lets the user know which hardware he has selected. Ensure the **text box's** ReadOnly property is set to True. (1 Mark: GUI).
    g. Write an **Event Handler** that highlights the entire row of the **DataGridView** control when any of a single row's cells are clicked (2 Mark: Functionality).

    h. Write a **Method** that selects all the data from the highlighted row of the **DataGridView** control and displays the **manufacturer field**, the **model field** and the **cost field** in the text box at the bottom of the form (2 Marks: Functionality).
    i. Add a **Cancel** button that terminates the application (1 Mark: GUI, 1 Mark: Functionality).
    j. Add a "**Next**" button that takes the user to the **ProductInfoForm** and hides **SelectForm**. This option will be grayed-out (disabled) until the user has selected a row from the **DataGridView** control indicating the computer hardware he has selected (1 Mark: GUI, 1 Mark: Functionality)
    k. Set the **ControlBox** properties of the form to False and the **StartPosition** properties of the form to **CenterScreen** (1 Mark: GUI).

3. **ProductInfoForm: (SubTotal: 6 Marks: GUI, 10 Marks: Functionality)**
    a. Add a **MenuStrip** to the form (see the **ProductInfoForm** Menu Strip diagram for the layout) (1 Mark: GUI).
    b. Add a series of **Labels** and **Buttons** separated by **GroupBoxes** (as appropriate) that display the computer component fields the user chose from **SelectForm** or a **previously saved file** (1 Mark: GUI, 2 Marks: Functionality)
    c. Write an **Event Handler** for the **Open** menu selection that invokes the **OpenFileDialog** box and allows the user to **load data** from a previously saved file. **Add a filter** to the dialog box to search for text files (*.txt) by default. (2 Marks: Functionality)
    d. Write an **Event Handler** for the **Save** menu selection that invokes the **SaveFileDialog** box and allows the user to **save his current hardware selection**. Indicate a **default filename** for the dialog box is "Product.txt" by default. (2 Marks: Functionality)
    e. Add a **Select Another Product** button that hides the **ProductInfoForm** and shows the **SelectForm**. When the user goes back to the **SelectForm**, his new selection is reflected on the **ProductInfoForm** when he returns. The **Select Another Product** button uses a **shared event Handler** with the **Select Another Product** Menu Selection (1 Mark: GUI, 2 Marks: Functionality).
    f. Add a **Cancel** button that terminates the application. The cancel button uses a **shared event Handler** with the **Exit** Menu Selection (1 Mark: GUI, 1 Mark: Functionality).
    g. Add a "**Next**" button that takes the user to the **OrderForm** and hides **SelectForm**. This option will be grayed-out (disabled) until the user has either made a selection from **SelectForm** or opened a previously saved file containing the user's selection. (1 Mark: GUI, 1 Mark: Functionality)
    h. Set the **ControlBox** properties of the form to False and the **StartPosition** properties of the form to **CenterScreen** (1 Mark: GUI).

4. **OrderForm: (SubTotal: 7 Marks: GUI, 11 Marks: Functionality)**
    a. Add a **MenuStrip** to the form (see the **OrderForm** Menu Strip diagram for the layout) (1 Mark: GUI).
    b. The **About Menu selection** will display an **AboutBox** containing the programmer's name, the version of the program and a website and contact number for the "Dollar Computers" (1 Mark: Functionality)
    c. Add a series of **Labels** and **Buttons** separated by **GroupBoxes** (as appropriate) that display the computer component fields the user chose from **SelectForm** or a **previously saved file** (1 Mark: GUI, 2 Marks: Functionality).
    d. Add a **PictureBox** that displays an image of the computer hardware that is selected from the project's Resource library (1 Mark: GUI, 1 Mark: Functionality).
    e. Add a separate **GroupBox** and appropriate labels that display the **cost** of the hardware, the **sales tax (at 13%)** and the **total cost** including tax (1 Mark: GUI, 1 Mark: Functionality).

f. Write an Event Handler for the **Print** menu selection that allows the user to display a MessageBox that tells the user that his selection is printing (1 Mark: Functionality).

g. Add a **Back** button that hides the **OrderForm** and shows the **ProductInfoForm**.
If the user goes back to the **SelectForm**, or loads a previously saved hardware configuration, his new selection is reflected on the **OrderForm** when he returns. The **Back** button uses a **shared event procedure** with the **Back** Menu Selection (1 Mark: GUI, 2 Marks: Functionality).

h. Add a **Cancel** button that terminates the application. The **Cancel** button uses a **shared event procedure** with the **Exit** Menu Selection (1 Mark: GUI, 1 Mark: Functionality).

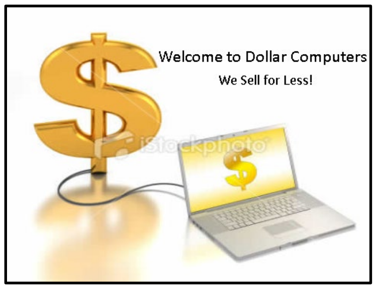
i. Add a "**Finish**" button that displays a **MessageBox** when clicked. The **MessageBox** will display a message to the user that thanks him for his business and advises him that his order will be processed in 7-10 business days. Ensure this information is double spaced and displayed over at least two lines. The program **terminates** when the user clicks the **OK button**.(2 Marks: Functionality).

j. Set the **ControlBox** properties of the form to False and the **StartPosition** properties of the form to **CenterScreen** (1 Mark: GUI).

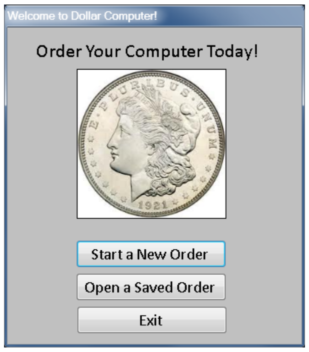
5. **Solution Structure (3 Marks: Program Structure):**
   a. Your solutions should include a Windows Forms named `StartForm.cs`, **SelectForm.cs**, `ProductInfoForm.cs`, and `OrderForm.cs` (1 Mark: Program Structure).
   b. The Windows Form and all attached **UI Controls** must have appropriate Variable names with the following format: **ControlNameUIControlType** (e.g. **CalculateBMIButton**) (1 Mark: Program Structure).
   c. Ensure all *private* class member variables (instance variables) use an underscore character (**_**) at the beginning of the identifier name to signify that they are private (or protected) (1 Mark: Program Structure).
6. Include **Internal Documentation** for your Application **(5 Marks: Internal Documentation):**
   a. Ensure you include a **comment header** for your **C# files** that indicate: the **App name**, **Author's name**, **Student ID**, **App Creation Date**, **App description** (2 Marks: Internal Documentation).
   b. Ensure you include **section headers** for all of your **Event Handlers, Classes,** and any **functions** (1 Marks: Internal Documentation)
   c. Ensure all your code uses **contextual variable names** that help make the files human-readable (1 Marks: Internal Documentation).
   d. Ensure you include **inline comments** that describe your GUI Design and Functionality. (1 Marks: Internal Documentation).
7. Share your files on **GitHub** to demonstrate Version Control Best Practices **(4 Marks: Version Control).**
   a. Your repository must include **your code** and be well structured (2 Marks: Version Control).
   b. Your repository must include **commits** that demonstrate the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

FINAL FORM (SAMPLES)

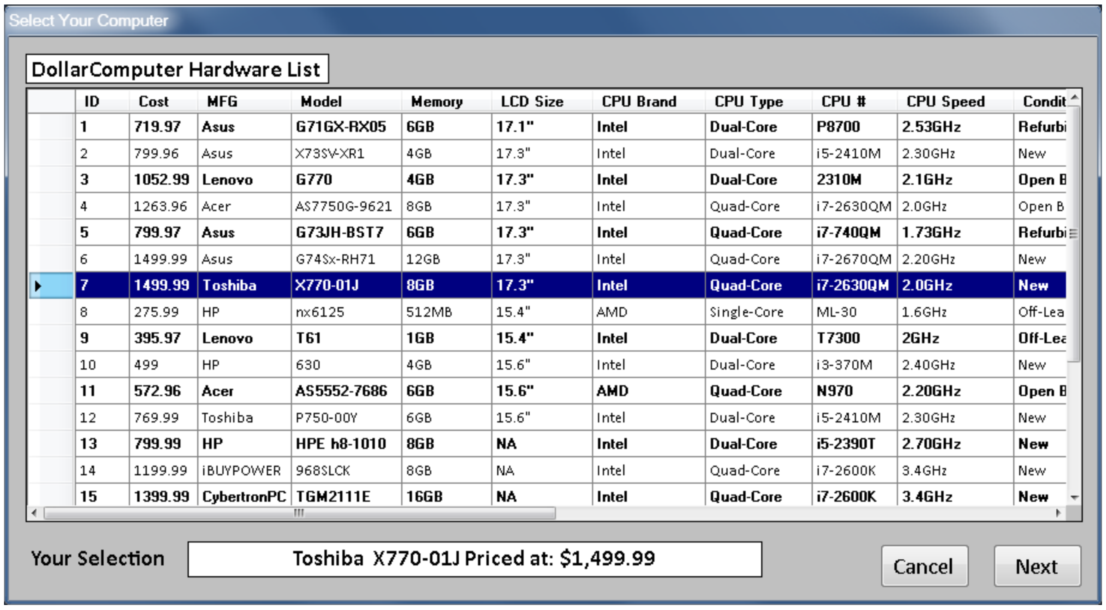
1. **SplashScreen**



2. **StartForm**



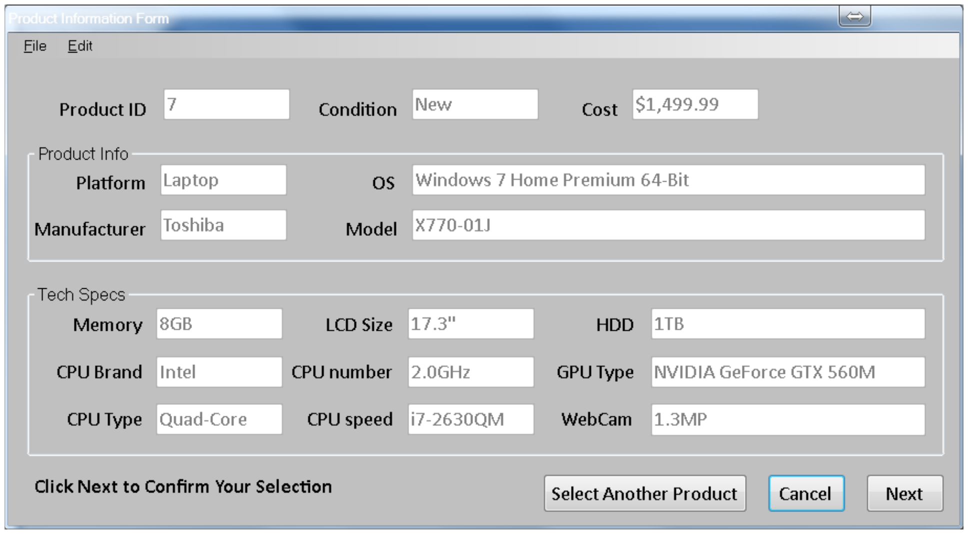
3. **SelectForm**



ID	Cost	MFG	Model	Memory	LCD Size	CPU Brand	CPU Type	CPU #	CPU Speed	Condit
1	719.97	Asus	G71GX-RX05	6GB	17.1"	Intel	Dual-Core	P8700	2.53GHz	Refurbi
2	799.96	Asus	X73SV-XR1	4GB	17.3"	Intel	Dual-Core	i5-2410M	2.30GHz	New
3	1052.99	Lenovo	G770	4GB	17.3"	Intel	Dual-Core	2310M	2.1GHz	Open B
4	1263.96	Acer	AS7750G-9621	8GB	17.3"	Intel	Quad-Core	i7-2630QM	2.0GHz	Open B
5	799.97	Asus	G73JH-BST7	6GB	17.3"	Intel	Quad-Core	i7-740QM	1.73GHz	Refurbi
6	1499.99	Asus	G74Sx-RH71	12GB	17.3"	Intel	Quad-Core	i7-2670QM	2.20GHz	New
7	1499.99	Toshiba	X770-01J	8GB	17.3"	Intel	Quad-Core	i7-2630QM	2.0GHz	New
8	275.99	HP	nx6125	512MB	15.4"	AMD	Single-Core	ML-30	1.6GHz	Off-Lea
9	395.97	Lenovo	T61	1GB	15.4"	Intel	Dual-Core	T7300	2GHz	Off-Lea
10	499	HP	630	4GB	15.6"	Intel	Dual-Core	i3-370M	2.40GHz	New
11	572.96	Acer	AS5552-7686	6GB	15.6"	AMD	Quad-Core	N970	2.20GHz	Open B
12	769.99	Toshiba	P750-00Y	6GB	15.6"	Intel	Dual-Core	i5-2410M	2.30GHz	New
13	799.99	HP	HPE h8-1010	8GB	NA	Intel	Dual-Core	i5-2390T	2.70GHz	New
14	1199.99	iBUYPOWER	968SLCK	8GB	NA	Intel	Quad-Core	i7-2600K	3.4GHz	New
15	1399.99	CybertronPC	TGM2111E	16GB	NA	Intel	Quad-Core	i7-2600K	3.4GHz	New

Your Selection: Toshiba X770-01J Priced at: $1,499.99

4. **ProductInfoForm**



Product ID	7	Condition	New	Cost	$1,499.99

Product Info

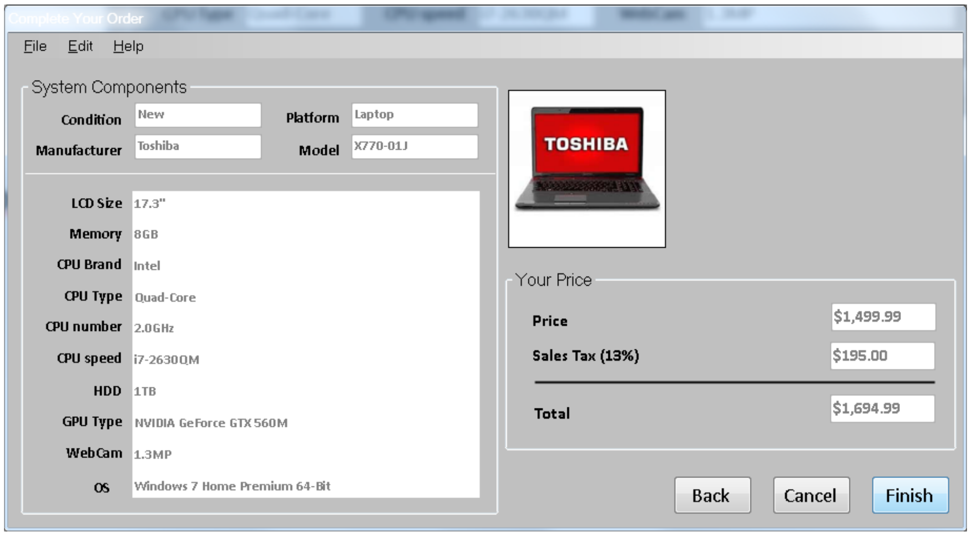
Platform	Laptop	OS	Windows 7 Home Premium 64-Bit
Manufacturer	Toshiba	Model	X770-01J

Tech Specs

Memory	8GB	LCD Size	17.3"	HDD	1TB
CPU Brand	Intel	CPU number	2.0GHz	GPU Type	NVIDIA GeForce GTX 560M
CPU Type	Quad-Core	CPU speed	i7-2630QM	WebCam	1.3MP

Click Next to Confirm Your Selection

5. **OrderForm**



System Components

Condition	New	Platform	Laptop
Manufacturer	Toshiba	Model	X770-01J

LCD Size	17.3"
Memory	8GB
CPU Brand	Intel
CPU Type	Quad-Core
CPU number	2.0GHz
CPU speed	i7-2630QM
HDD	1TB
GPU Type	NVIDIA GeForce GTX 560M
WebCam	1.3MP
OS	Windows 7 Home Premium 64-Bit

Your Price

Price	$1,499.99
Sales Tax (13%)	$195.00
Total	$1,694.99

SUBMITTING YOUR WORK

Your submission should include:
1. A zip archive of your project files uploaded to eCentennial
2. A link to your project files on GitHub.

EVALUATION

Feature	Description	Marks
GUI / Interface Design	UI Controls meet the application requirements. Display elements are deployed in an attractive manner. Appropriate contrast is applied to application UI Controls and any background colours applied so that all text is legible.	27
Functionality	The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program.	38
Program Structure	Your main "driver" class is named Program and it creates objects that are defined in other classes. All other classes are contained in their own files. Your classes use **public** properties and related **private** member variables wherever possible.	3
Internal Documentation	A program header is present and includes the name of the program, the name of the student, student number, date last modified, a short revision history and a short description of the program. All methods and classes include headers that describe their functionality and scope and follow commenting best practices. Inline comments are used to indicate code function where appropriate. Variable names are contextual wherever possible.	5
Version Control	GitHub commit history demonstrating regular updates.	4
Total		**77**

This assignment is weighted **10%** of your total mark for this course.

All Assignments are due at the beginning of class.
Late submissions:
- 20% deducted for each additional day.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:
1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 25% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.