Hints for Project Five

Separate User Interface Logic from Business Logic

One of the most important things to get right in this project is the separation of user interface logic from the business logic of the program. To make sure that you are doing this correctly, follow these two rules:

1. There should be no code in your Form class that does business logic, for example, you should not try computing the sales tax on a purchase in any of the methods in your Form class.
2. There should be no code in your SalesInvoice class that tries to interact with GUI components on the Form. For example, do not try to get data directly from a TextBox in any of the methods in your SalesInvoice class.

Creating the Sales Ticket

One of the things that code in your Form class will have to do is to create the Sales Ticket. Remember that anything having to do with the user interface is work that has to be done by the Form. To create a MessageBox, you call the *Show( )* method in the MessageBox class. The Show( ) method takes one parameter, which is the string that you want displayed in the MessageBox. It takes a little bit of work to create this string. The easiest way to do this is to create a different string for each line that you want to display in the MessageBox. This is done using the *Format( )* method in the String class. The statement in the line below takes an integer variable named *theUnits* and uses the format specification provided to build a string that you can then display in the message box.

```
  string s1 = string.Format("Quantity: {0:d} units\n",
mySale.theUnits );
```

Suppose that you now have three string objects, *s1, s2,* and *s3* that you want displayed in the MessageBox. You can do this by concatenting the strings together like this:

```
  MessageBox.Show(s1 + s2 + s3);
```