CIS 400 Exam II

Name: Robert Clancy WID: 842664019 Section: B

Vocabulary

The following questions deal with object-orientation on the theoretical level, not a specific language

Q1. What are the roles of the event loop and event queue?

The event loop constantly checks for new events raised by the user. These events are then placed in an event queue, where they are progressively executed by the program.

Confidence (circle one): very confident - (5) [✓ (4)] (3) (2) (1) - Not confident at all

Q2. Describe data binding:

Databinding is a concept in User Interface design where two variables holding data are connected together so that the View and Model work synchronously to user input.

Confidence (circle one): very confident - (5) [✓ (4)] (3) (2) (1) - Not confident at all

Q3. What is composition?

Composition is when an object is made up by a collection of other objects, Our Combo.cs class in BleakwindBuffet is an example of a composition

Confidence (circle one): very confident - [✓ (5)] (4) (3) (2) (1) - Not confident at all

C# Keywords

The following questions deal specifically with the C# language

Q4. What is the "elements tree"?

It is a a tree-like structure that makes it easier for programmers to interact with specific components in the tree, by being able to move up (by getting the current element's parent) or down (by getting the current element's child) the tree.

Confidence (circle one): very confident - (5) [✓ (4)] (3) (2) (1) - Not confident at all

Q5. What is the "as" keyword used for?

The "as" keyword is an example of Syntactic sugar in C#. It is an easy and clean way to typecast an object to another compatible type.

Confidence (circle one): very confident - [✓ (5)] (4) (3) (2) (1) - Not confident at all

Q6. What does the PropertyChanged event represent?

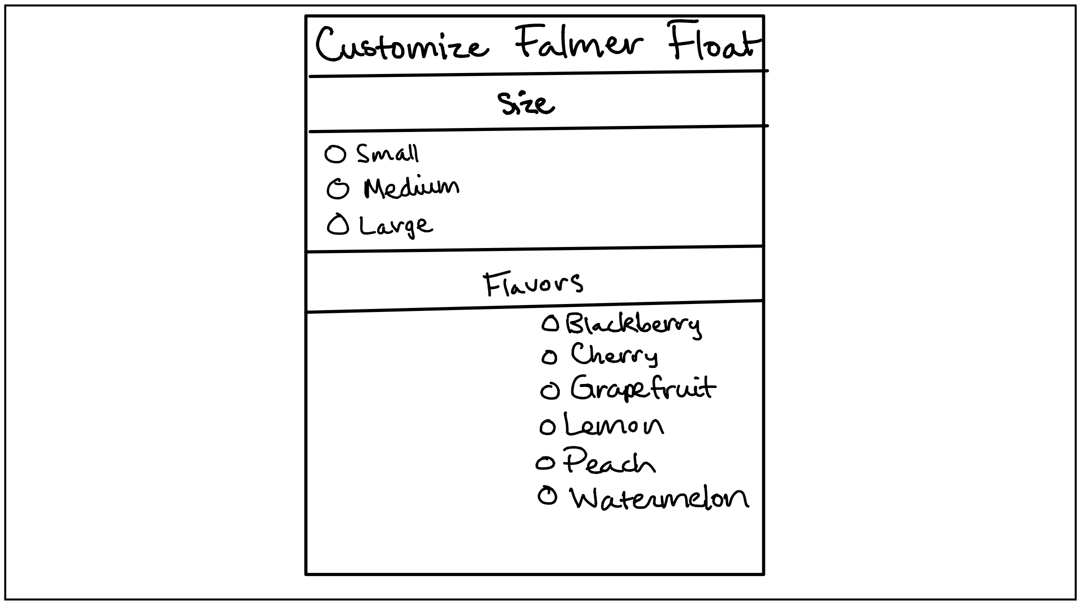
PropertyChanged is an event that is executed (when implemented correctly) that dynamically updates the binded elements in the View when the property it is binded to changes.

Confidence (circle one): very confident - [✓ (5)] (4) (3) (2) (1) - Not confident at all

Coding Questions

These questions use the diagrams, code listings, and other details found on the last page of the exam.

Q7. Draw the CustomizeFalmerFloat UserControl defined in the XAML as it will appear on-screen:



Confidence (circle one): very confident - (5) (4) (3) [✓ (2)] (1) - Not confident at all

Q8. Write a Design Template that displays a bound FalmerFloat's Name, Price, and Calories:

`<DesignTemplate>` Not familiar with this term......

```
< StackPanel >
    < TextBlock  content="{Binding: Name}"/>
    < TextBlock  content= {Binding: Price}"/>
    < TextBlock  content= {Binding: Calories}"/>
</StackPanel
```

`</DesignTemplate>`

Confidence (circle one): very confident - (5) (4) (3) (2) [✓ (1)] - Not confident at al

Q9. Complete the unit tests for the Famler Float:

```
public class FalmerFloatTests
{
    [Theory]
    [InlineData(Size.Small, 3.00)]
    [InlineData(Size.Medium, 3.50)]
    [InlineData(Size.Large, 4.00)]
    public void HasCorrectPriceForSize(Size size, double price)
    {
              Falmer Float ff = new Falmer Foat () { Size = size };
              Assert. Equal ( ff.Price, price );
    }

    [Theory]
[ InlineData ( Flavor. Blackberry ) ]
[ InlineData ( Flavor. Cherry     ) ]
[ InlineData ( Flavor. Grapefruit ) ]
[ InlineData ( Flavor. Lemon      ) ]
[ InlineData ( Flavor. Peach      ) ]
[ InlineData ( Flavor. Watermelon) ]
    public void CanSetFlavor(SodaFlavor flavor)
    {
        var ff = new FalmerFloat();
        Assert.Equal(flavor, ff.Flavor);
    }

    [Theory]
    [InlineData(Size.Small, SodaFlavor.Blackberry)]
    [InlineData(Size.Small, SodaFlavor.Lemon)]
    [InlineData(Size.Medium, SodaFlavor.Watermellon)]
    [InlineData(Size.Medium, SodaFlavor.Cherry)]
    [InlineData(Size.Large, SodaFlavor.Peach)]
    [InlineData(Size.Large, SodaFlavor.Grapefruit)]
    public void NameIsCorrectForSizeAndFlavor(Size size, SodaFlavor flavor)
    {
          var ff = new FalmerFloat () { Size = size, Flavor = flavor };
          string name= size.ToString() + flavor.ToString() + "Falmer Float";
          Assert.Equal( name, ff.ToString() );
    }

    [Theory]
    [InlineData(SodaFlavor.Blackberry)]
    [InlineData(SodaFlavor.Cherry)]
    [InlineData(SodaFlavor.Grapefruit)]
    public void ChangingFlavorNotifiesOfNameChange(SodaFlavor flavor)
    {
           var ff= new FalmerFloat () { Flavor = SodaFlavor.Lemon };
            Assert. PropertyChanged ( ff , "Name") =>
                   ff. Flavor = flavor )};
    }
}
```

Confidence (circle one): very confident - (5) [✓ (4)] (3) (2) (1) - Not confident at all

Q10. Write the code corresponding to the FalmerFloat class defined in the UML and values table. You may add any private or protected fields, properties, or methods you feel are needed, but any public fields, properties, and methods should match the UML specification:

```
public string Name {
    get => ToString();
}

namespace Data
{
public class FalmerFloat : IMenuItem, INotifyPropertyChanged
{
    public PropertyChangedHandler PropertyChanged;
    public double Price {
        get {
            if ( Size = Size.Small) return 3.00;
            else if ( Size = Size.Medium) return 3.50;
            else return 4.00;
        }
    }
    public uint Calories {
        get {
            if ( Size = Size.Small) return 380;
            else if ( Size = Size.Medium) return 468;
            else return 575;
        }
    }
    private Size size;
    publice Size Size {
        get => size
        set {
            size = value.
            NotifyChanged?.Invoke(this, new PropertyChanged(" Size ")
            NotifyChanged?.Invoke(this, new PropertyChanged(" Price ")
            NotifyChanged?.Invoke(this, new PropertyChanged(" Calories ")
            NotifyChanged?.Invoke(this, new PropertyChanged(" Name ")
        }
    }
    private SodaFlavor flavor;
    publice SodaFlavor Flavor;
        get => flavor
        set {
            flavor = value;
            NotifyChanged?.Invoke(this, new PropertyChanged(" Size ")
        }
    }
    public override string ToString() {
        return "{Flavor} {Size} + FalmerFloat";
    }
}
```

Confidence (circle one): very confident - (5) [✓ (4)] (3) (2) (1) - Not confident at all

Diagrams and Code Listings

This information applies to questions 7-10

```
<UserControl x:Class="PointOfSale.CustomizeFalmerFloat"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:PointOfSale"
             xmlns:data="clr-namespace:Data;assembly=data"
             xmlns:system="clr-namespace:System;assembly=mscorlib"
             mc:Ignorable="d"
             d:DesignHeight="450" d:DesignWidth="800">
    <UserControl.Resources>
        <ObjectDataProvider x:Key="flavors"
                            MethodName="GetValues"
                            ObjectType="{x:Type system:Enum}">
            <ObjectDataProvider.MethodParameters>
                <x:Type TypeName="data:SodaFlavor"/>
            </ObjectDataProvider.MethodParameters>
        </ObjectDataProvider>
        <ObjectDataProvider x:Key="sizes"
                            MethodName="GetValues"
                            ObjectType="{x:Type system:Enum}">
            <ObjectDataProvider.MethodParameters>
                <x:Type TypeName="data:Size"/>
            </ObjectDataProvider.MethodParameters>
        </ObjectDataProvider>
        <Style TargetType="ListViewItem">
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate>
                        <RadioButton
                            Content="{TemplateBinding ContentPresenter.Content}"
                            IsChecked="{Binding Path=IsSelected,
                                RelativeSource={RelativeSource TemplatedParent},
                                Mode=TwoWay}"/>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </UserControl.Resources>
    <StackPanel>
        <TextBlock Text="Customize Falmer Float" FontSize="32"/>
        <TextBlock Text="Size" FontSize="20"/>
        <ListView Grid.Column="0"
                  ItemsSource="{Binding Source={StaticResource sizes}}"/>
        <TextBlock Text="Flavor" FontSize="20"/>
        <ListView Grid.Column="1"
                  ItemsSource="{Binding Source={StaticResource flavors}}"/>
    </StackPanel>
</UserControl>
```

Figure 1 - Falmer Float UML

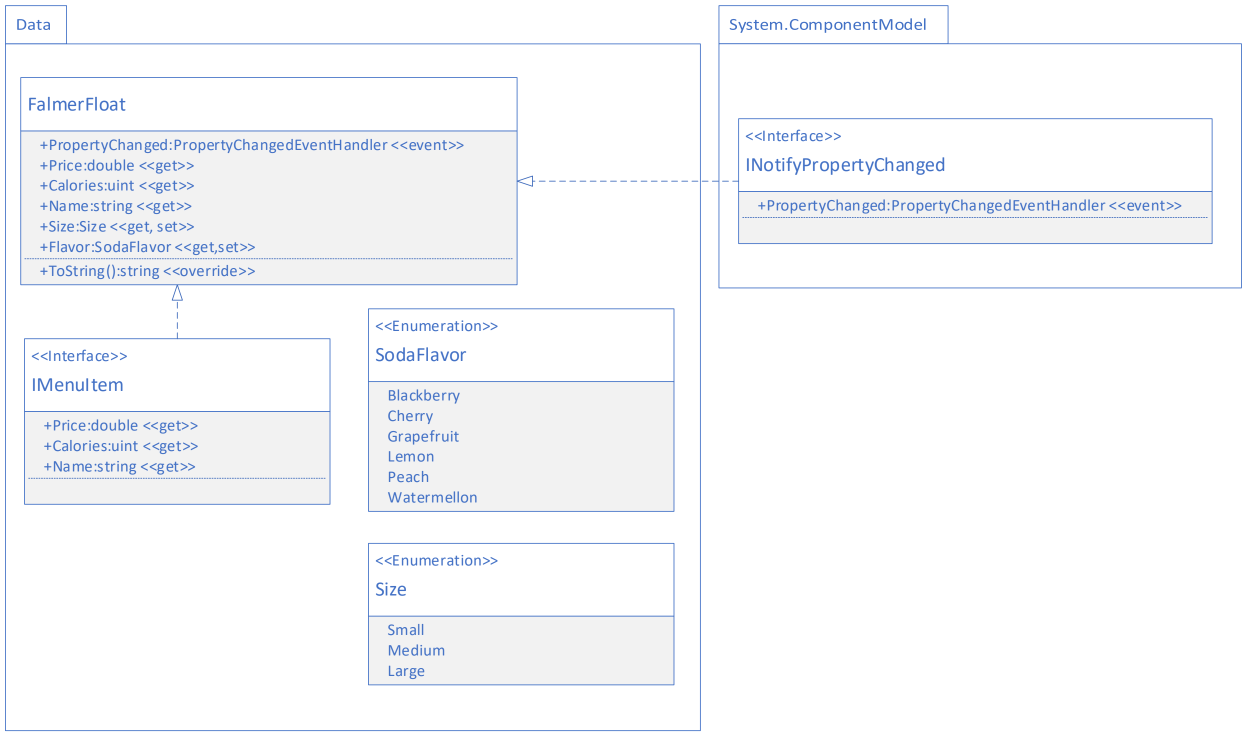


Table 1 - Falmer Float Values

Size	**Price**	**Calories**	**Name & ToString() ***
Small	3.00	380	Small [Flavor] Falmer Float
Medium	3.50	478	Medium [Flavor] Falmer Float
Large	4.00	565	Large [Flavor] Falmer Float

* [Flavor] should be the name of the SodaFlavor used in the float