WPF DQs.

S 1 Day 1

**1.** A Page has to be created with the following specifications:  
  
i. The title bar of the browser is "ExpenseIt".  
ii. The width of the browser is 550 device-independent pixels.  
iii.The height of the browser is 380 device-independent pixels.  
iv.The title of the page is "ExpenseIt - Home".  
  
What is the correct XMAL code snippet for the same?

**1)**. <Window   
       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
       x:Class="ExpenseIt.HomePage"  
       WindowTitle="ExpenseIt"  
       Title="ExpenseIt - Home"   
       WindowWidth="550" WindowHeight="380">  
 </Window>

**2)**. <Page   
       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
       x:Class="ExpenseIt.HomePage"  
       WindowTitle="ExpenseIt - Home"  
       Title="ExpenseIt"   
       WindowWidth="550" WindowHeight="380">  
</Form>

**3)**. **<Page   
       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
       x:Class="ExpenseIt.HomePage"  
       WindowTitle="ExpenseIt"  
       Title="ExpenseIt - Home"   
       WindowWidth="550" WindowHeight="380">  
     </Page>**

**4)**.      None of these.

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### 2.In XAML, you can turn any attribute (property on the class) into a child element by using the following pattern:

**1)**. **ElementName.AttributeName**

**2)**. ElementName:AttributeName

**3)**. **<Element Attribute="value">**

**4)**. None of these

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### 3. Q. Look at the following XAML code. <Button MinWidth="75" Margin="10"> <Button.Style> <Style TargetType="{x:Type Button}"> <Style.Triggers> <Trigger Property="IsMouseOver" Value="True"> <Setter Property="Foreground" Value="Blue"/> </Trigger> </Style.Triggers> </Style> </Button.Style> OK </Button> What is the incorrect statement with respect to the above code?

**1)**. The Property Trigger attaches the MouseOver Event with the Button.

**2)**. When the Mouse is hovered over the Button, The foreground color of the Button will Change to Blue.

**3)**. **We will have to define another Trigger to Change the color back to original when the mouse leaves the Button**

**4)**. **With Property triggers, we do not need to write any other script when the associated event takes place.**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

##### 4. Refer the XAML markup file called App.xaml. <Application    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"   StartupUri="Login.xaml"> </Application> What is this code snippet doing?

**1)**. **This file defines a WPF application, and is also used to specify the UI  (Login.xaml) to automatically   
show when the application starts.**

**2)**. This file defines a WPF Page, with the name Login.xaml as a startup Page.

**3)**. This file defines a WPF application, and is also used to create the UI (Login.xaml)

**4)**. None of these

**Solution** :  
option [1] is correct

**Attempted** :  
option [1] is attempted

##### 5. . <Button> Submit Data <Button> The "Submit Data" in the above code is referred as?

**1)**. Text

**2)**. **Content**

**3)**. Caption

**4)**. None of these

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### 6. Select the correct XAML code snippet to Add a Button on a Window with a StackPanel Layout.

**1)**. StackPanel>  
       <Button ="Click Me"/>  
</StackPanel>

**2)**. <StackPanel>  
       <Button Text="Click Me"/>  
 </StackPanel>

**3)**. **<StackPanel>  
       <Button Content="Click Me"/>  
 </StackPanel>**

**4)**. <Panel>  
       <Button Content="Click Me"/>  
</Panel>

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### 7. Look at the following XAML code. <Grid>   <Grid.Resources>     <LinearGradientBrush x:Key="MyBrush">       <GradientStop Color="Green" Offset="0" />       <GradientStop Color="Red" Offset="1" />     </LinearGradientBrush>   </Grid.Resources> </Grid> If we want to add a Button on the Grid, having Background property set to the markup extension  defined in the above code, what would be the correct way to do this?

**1)**. <Button Background="{MyBrush}">Click Me</Button>

**2)**. <Button StaticResource ="{ Background MyBrush}">Click Me</Button>

**3)**. **<Button Background="{StaticResource MyBrush}">Click Me</Button>**

**4)**. None of the above

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### 8. The .NET Framework 3.0 \_\_\_\_\_\_\_\_\_is the WPF version of an ASP.NET web application,  with a limited amount of WPF namespaces and functionality available to it, because of  the browser's security access limitations on the client.

**1)**. Windows Application

**2)**. Service Library Application

**3)**. **XAML Browser Application (XBAP)**

**4)**. None of the above

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### 9. Refer the code below: <Grid Margin="10">       <Grid.ColumnDefinitions>         <ColumnDefinition />       </Grid.ColumnDefinitions>       <Grid.RowDefinitions>         <RowDefinition Height="Auto" />         <RowDefinition />         <RowDefinition Height="Auto" />       </Grid.RowDefinitions>     </Grid> Select the statement which is not true with respect to the above code.

**1)**. The above XAML markup creates a single-column table

**2)**. **The table has one row and a 10-pixel margin**

**3)**. The table is added to the Grid

**4)**. This table within the Grid Layout can be added to a Page.

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### 10. Select the correct code snippet to create a Button that has a red text and a Blue Background,  as well as text specified as Content:

**1)**. **<Button Background="Blue" Foreground="Red" Content="This is a button"/>**

**2)**. <Button Background="Red" Foreground="Blue" Content="This is a button"/>

**3)**. **<Button Backcolor="Blue" Forecolor="Red" Content="This is a button"/>**

**4)**. None of these

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

WPF S1 Day2

1. Match the following w.r.t. Binding Modes in Data Binding:  
     
   Set A  
   One Time  
   One Way  
   Two Way  
     
   Set  B  
   a. Changes to the source object can also propagate to the target  
   b. Updates the target property when the binding is created  
   c. Updates either the target or the source object when either change
2. **1)**. **1-b, 2-a, 3-c**
3. **2)**. 1-a, 2-b, 3-c
4. **3)**. 1-c, 2-b, 3-a
5. **4)**. 1-b, 2-c, 3-a
6. **Solution** :  
   option [1] is correct
7. **Attempted** :  
   option [1] is attempted

##### 2.Which WPF feature allows you to sort the collection when an item in a collection is updated at runtime?

**1)**. ObservableCollection

**2)**. **Live Shaping**

**3)**. INotifyPropertyChanged

**4)**. **Data Binding**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### 2.Which WPF feature allows you to sort the collection when an item in a collection is updated at runtime?

**1)**. ObservableCollection

**2)**. **Live Shaping**

**3)**. INotifyPropertyChanged

**4)**. **Data Binding**

**Solution** :  
option [2] is correct

**Attempted** :  
option [4] is attempted

##### 3. Select which statements are true with respect to Routed Events:

**1)**. **Routed events enable an element to handle an event that was raised by a different element,   
as long as the elements are connected through an element tree relationship.**

**2)**. **When specifying event handling with a XAML attribute, the routed event can be listened for and handled   
on any element, including elements that do not list that particular event in the class members table.**

**3)**. **A particular event feature that is unique and fundamental to WPF is a routed event, which is available   
in .NET Framework 2.0.**

**4)**. None of the above statements are True.

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,3] are attempted

##### 4.  On a Window, a Button is added with the following code:   <Button Click="ClickHandler" >Click Me!</Button> The event name and the name of the handler to use are specified in the markup. Identify the correct code snippet that implements event handler in the code-behind to change the  background color of the Button.

**1)**. public partial class MyPageCode  
       {  
         void ClickHandler(object sender, RoutedEventArgs e)  
         {  
           Button b = new Button;  
           b.Background = Brushes.Red;  
         }  
       }

**2)**. **public partial class MyPageCode  
       {  
         void ClickHandler(object sender, RoutedEventArgs e)  
         {  
            sender.Background = Brushes.Red;  
         }  
       }**

**3)**. **public partial class MyPageCode  
       {  
         void ClickHandler(object sender, RoutedEventArgs e)  
         {  
           Button b = e.Source as Button;  
           b.Background = Brushes.Red;  
         }  
       }**

**4)**. None of the above

**Solution** :  
option [3] is correct

**Attempted** :  
option [2] is attempted

##### 5.Rajeev wants to define a Border style for a Page as a keyed Resource in a resource dictionary.  Which is the correct code snippet to achieve the same?

**1)**. <Window.Resources>  
       <SolidColorBrush x:Key="MyBrush" Color="Gold"/>  
       <Style TargetType="Bordercolor" x:Key="PageBackground">  
         <Setter Property="Background" Value="Blue"/>  
       </Style>  
     ...  
  
     </Window.Resources>

**2)**. **<Page.Resources>  
       <SolidColorBrush x:Key="MyBrush" Color="Gold"/>  
       <Style TargetType="Border" x:Key="PageBackground">  
         <Setter Property="Background" Value="Blue"/>  
       </Style>  
     ...  
  
     </Page.Resources>**

**3)**. <Page.Resources>  
       <SolidColorBrush Key="MyBrush" Color="Gold"/>  
       <Style TargetType="BorderColor" Key="PageBackground">  
    <Setter Property="Background" Value="Blue"/>  
       </Style>  
     ...  
  
     </Page.Resources>

**4)**. <Page.Resources>  
       <SolidColorBrush Key="MyBrush" Color="Gold"/>  
       <Style TargetType="Border" Key="PageBackground">  
         <Setter Property="Background" Value="Blue"/>  
       </Style>  
     ...  
  
</Page.Resources>

**Solution** :  
option [2] is correct

**Attempted** :  
No options are Attempted

##### 6. Which of the following are correct w.r.t. Routed Events?

**1)**. **It allows a click that begins in a button to rise up to that button's container and then to the containing page.**

**2)**. **PreviewMouseLeftButtonDown is a Bubbling event**

**3)**. **It allows you to centralize the event handling logic**

**4)**. **WPF supports bubbling and Tunelling of events**

**Solution** :  
option [1,3,4] are correct

**Attempted** :  
option [2,4] are attempted

1. With the Binding class, you can configure a number of binding options.   
   Match the following options of Binding class with respect to their appropriate usage.  
     
   Set A  
   1. Source  
   2. RelativeSource  
   3. ElementName  
   4. Path  
   5. XPath  
   6. Mode  
     
   Set B  
   a.With RelativeSource, you can specify the source in relation to the target object.   
      This is useful to display error messages when the source of the error comes from the same control.   
   b. The mode defines the direction for the binding.  
   c. With the Source property, you define the source object for data binding.  
   d.If the source is a WPF element, you can specify the source with the ElementName property.   
   e. With an XML data source, you can define an XPath query expression to get the data for binding.   
   f. With the Path property, you specify the path to the source object.
2. **1)**. **1-c, 2-a, 3-d, 4-f, 5-e, 6-b**

**2)**. 1-a, 2-c, 3-d, 4-f, 5-e, 6-b

**3)**. 1-a, 2-c, 3-d, 4-e, 5-f, 6-b

**4)**. 1-c, 2-a, 3-b, 4-e, 5-f, 6-d

**Solution** :  
option [1] is correct

**Attempted** :

Select which statements are True w.r.t. the code snippet.  
  
<StackPanel>  
  <Border Style="{StaticResource PageBackground}">  
...  
  </Border>  
</StackPanel>

**1)**. **The above example sets the value of the Style property using attribute syntax**

**2)**. The above example sets the value of the Style property using Property Element syntax

**3)**. **The attribute references a particular markup extension, StaticResource, which returns a reference   
to a style that was previously instantiated as a keyed resource in a resource dictionary**

**4)**. This is an invalid syntax, as you can not refer the StaticResource markup extension with Style.

**Solution** :  
option [1,3] are correct

**Attempted** :  
No options are Attempted

The \_\_\_\_ property added to Binding markup extension can be used to specify an amount of time to pass   
after the property changes on the target before the source updates.

**1)**. ElementName

**2)**. Path

**3)**. DataContext

**4)**. **Delay**

**Solution** :  
option [4] is correct

**Attempted** :  
No options are Attempted

Which of the following interface enables binding to objects, the structure of   
which is not known until runtime?

**1)**. IComponent

**2)**. ICustomProvider

**3)**. INotifyPropertyChanged

**4)**. **ICustomTypeProvider**

**Solution** :  
option [4] is correct

**Attempted** :  
No options are Attempted

WPF S2 Day 1

Which of the following layout is not available in WPF?

**1)**. StackPanel

**2)**. Grid

**3)**. Canvas

**4)**. WrapPanel

**5)**. **Table**

**Solution** :  
option [5] is correct

**Attempted** :  
option [5] is attempted

WPF separates the Look and Feel of an user interface from its behavior.   
The Look and Feel is generally specified in the \_\_\_\_\_\_\_.

**1)**. HTML

**2)**. **XAML**

**3)**. XSLT

**4)**. XML

**Solution** :  
option [2] is correct

**Attempted** :  
option [2] is attempted

##### . \_\_\_\_\_\_\_\_\_ creates Top level elements StackPanel, Grid, Controls, Window etc.

### **Attempted**

1. System.Windows.Forms

### **Solution**

1. PresentationFramework.dll

2. System.Windows.Forms

3. WindowBase.dll

4. PresentationCore.dll

Which property of Dockpanel can be used as Attached property for child elements?

**1)**. ID

**2)**. Name

**3)**. **Dock**

**4)**. Position

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

\_\_\_\_\_\_\_\_ Translates the WPF higher level objects like layout panels, buttons into textures   
that Direct3D expects.

**1)**. PresentationFramework.dll

**2)**. System.Windows.Forms

**3)**. WindowBase.dll

**4)**. **Milcore.dll**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

You have to add a Button control and a StatusBar control to a WPF window.   
You need to ensure that the two controls meet the following requirements:  
  
- The StatusBar control sticks to the lower edge of the window.  
- The Button control has the minimum required size to display its content.  
  
Which of the following code snippet goes with the requirement?

**1)**. **<DockPanel>  
<StatusBar DockPanel.Dock="Bottom">Status Info</StatusBar>  
<Button>ClickMe</Button>  
</DockPanel>**

**2)**. <WrapPanel Orientation="Vertical">  
<Button>ClickMe</Button>  
<StatusBar>Status Info </StatusBar>  
</WrapPanel>

**3)**. **<DockPanel LastChildFill="False" >  
<WrapPanel DockPanel.Dock="Top" >  
<Button>OK</Button>  
</WrapPanel>  
<StatusBar DockPanel.Dock="Bottom">Information</StatusBar>  
</DockPanel>**

**4)**. <StackPanel>  
<Button>ClickMe</Button>  
<StatusBar>Status Info </StatusBar>  
</StackPanel>

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

We are creating a WPF application for our client where we have to put elements   
which stretch horizontally to occupy the entire width but not vertically.   
Which layout control we use for particular scenario?

**1)**. Grid Control

**2)**. Canvas Control

**3)**. **WrapPanel**

**4)**. StackPanel

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

We are creating a WPF application for our client where we have to put elements   
which stretch horizontally to occupy the entire width but not vertically.   
Which layout control we use for particular scenario?

**1)**. Grid Control

**2)**. Canvas Control

**3)**. **WrapPanel**

**4)**. StackPanel

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

We are creating a WPF application for our client where we have to put elements   
which stretch horizontally to occupy the entire width but not vertically.   
Which layout control we use for particular scenario?

**1)**. Grid Control

**2)**. Canvas Control

**3)**. **WrapPanel**

**4)**. StackPanel

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

**Q.**If you want to draw shapes on desired position of wpf window, which container control   
you should prefer?

**1)**. StackPanel

**2)**. Grid

**3)**. **Canvas**

**4)**. DockPanel

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

What is true about WPF ?

**1)**. **Its an entirely new graphical system for Windows**

**2)**. Its Support only 3D graphics

**3)**. **It doesn't support Media elements**

**4)**. **It is a Hardware accelerated design**

**Solution** :  
option [1,4] are correct

**Attempted** :  
option [1,3] are attempted

WPF S2 Day2

You are developing a Windows Presentation Foundation (WPF) application with multiple windows.  
The majority of the buttons within the application are styled consistently.   
However, three buttons within the application must be styled differently.  
You need to ensure that the application can handle this styling requirement.  
What should you do?

**1)**. Create the styles in the application resources. Specify the TargetType property to be Button   
for the default style but not for the three buttons.

**2)**. Create the styles in the main window resources. Specify the TargetType property to be Button   
for the default style but not for the three buttons.

**3)**. Create the styles in the main window resources. Omit the Key property for the default Button style   
and specify the Key property for the Button style and the three buttons.

**4)**. **Create the styles in the application resources. Omit the Key property for the default Button style   
and specify the Key property for the Button style and the three buttons.**

**Solution** :  
option [4] is correct

**Attempted** :  
option [4] is attempted

We are developing a WPF application in which we are dealing with custom collection.  
If there are any change in custom collection we have to send some notification to our   
UI elements which is bound with Collection.   
  
What should be used in the code to achieve this behavior?

**1)**. ICollectionChanged

**2)**. INotifyChanged

**3)**. **INotifyCollectionChanged**

**4)**. INotifyPropertyChanged

**5)**. All of the above

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

You are developing a Windows Presentation Foundation (WPF) application.   
You have a class named Orders, which contains OrderId and ShipOn properties.   
You place a control in the MainWindow.xaml file, as follows.  
  
<ComboBox x:Name="cmbOrders"           
            ItemSource="{Binding Orders}">  
</ComboBox>  
  
When you run the application, the ComboBox control displays the class name for every row.  
You need to ensure that the ComboBox control displays the OrderId and ShipOn values in columns.  
Which markup segment should you add at line 03?

**1)**. **<ItemsControl.ItemsPanel>  
<ItemsPanelTemplate>  
<Grid>  
<Grid.ColumnDefinitions>  
<ColumnDefinition />  
<CoIumndefinltion />  
</Grid.ColumnDefinitions>  
<TextBlock Grid.Column-"0" Text="{Binding ShipOn}"/>  
<TextBlock Grid.Column"0" Text="{Binding OrderId}"/> </Grid>  
</ItemsPanelTemplate>  
</ItemsControl.ItemsPanel>**

**2)**. <ItemsControl.ItemTempIate>  
<DataTemplate>  
<Grid>  
<TextBlock Text="{Binding OrderId}"/>  
<TextBlock Text="{Binding ShipOn}"/>  
</Grid>  
</DataTemplate>  
</ItemsControl.ItemTemplate

**3)**. **<ItemsControl.Itemlempiate>  
<DataTempIate>  
<Grid>  
<Grid.ColumnDefinitions>  
<ColumnDefinition />  
<ColumnDefinicion />  
</Grid.ColumnDefinitions>  
<TextBlock Grid.Coxumn"0" Text="{Binding OrderId}"/>  
<TextBlock Grid.Coxumn"1" Text="{Binding ShipOn}"/> </Grid>  
</DataTemplate>  
</ItemsControl.ItemTemplate>**

**4)**. <ItemsControl.ItemsPanel>  
<ItemsPanelTemplate>  
<Grid>  
<TextBlock Text ="{Binding OrderId}/ >  
<TextBlock Text ="{Binding ShipOn}/ >  
</Grid>  
</ItemsPanelTemplate>  
</ItemsControl.ItemPanel>

**Solution** :  
option [3] is correct

**Attempted** :  
option [1] is attempted

##### You are developing a Windows Presentation Foundation (WPF) application to display  Categories in ListBox. You have created category class as follows: public class Category {       public int ID {get; set; }       public string Name { get; set; } } On Window load event you have written following code : List<Category> categoryList = new List<Category>(); categoryList.Add(new Category(){ID=101, Name="Accessories"); categoryList.Add(new Category(){ID=102, Name="Bikes"); categoryList.Add(new Category(){ID=103, Name="Cars"); lstCategory.DataContext = categoryList; Which of the following is the correct representation for the ListBox lstCategory,  so that it will display categories?

**1)**. <ListBox name=lstCategory></ListBox>

**2)**. <ListBox name=lstCategory Binding="Category"></ListBox>

**3)**. **<ListBox name=lstCategory ItemSource={Binding}></ListBox>**

**4)**. <ListBox name=lstCategory DataSource={Binding}></ListBox>

**Solution** :  
option [3] is correct

**Attempted** :  
option [3] is attempted

##### Which statements are True about Triggers in WPF ? 1)Triggers get active when a property lost a specified value. 2)Triggers get active when a specified event is fired. 3)Triggers get active when a binding expression reaches a specified value.

**1)**. 1

**2)**. 2

**3)**. 3

**4)**. 1 & 3 Both

**5)**. **2 & 3 Both**

**Solution** :  
option [5] is correct

**Attempted** :  
option [5] is attempted

Which of the statement is not true about CLR Property vs. Dependency Property?

**1)**. **Both types of properties are used to initialize and retrieve values of data members**

**2)**. Dependency Property can apply Styles which cannot be done with CLR property

**3)**. **Both Types of properties are used for databinding**

**4)**. All the above statements are false

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

##### Which of the statement is not true about CLR Property vs. Dependency Property?

**1)**. **Both types of properties are used to initialize and retrieve values of data members**

**2)**. Dependency Property can apply Styles which cannot be done with CLR property

**3)**. **Both Types of properties are used for databinding**

**4)**. All the above statements are false

**Solution** :  
option [1] is correct

**Attempted** :  
option [3] is attempted

 To apply styles for Label \_\_\_\_\_\_\_\_\_ or \_\_\_\_\_\_\_\_\_ should be specified.

**1)**. **Style="{DynamicResource Stylename}"**

**2)**. **Style="{StaticResource Stylename}"**

**3)**. Style="{StaticResource: Stylename}"

**4)**. All of the above

**Solution** :  
option [1,2] are correct

**Attempted** :  
option [1,2] are attempted

In our WPF application we want to go for Event Tunneling implementation but confused with   
available snippets. Can you suggest which one we pick for implementation?

**1)**. Button1\_Click

**2)**. MouseDoubleClick

**3)**. **PreviewMouseDoubleClick**

**4)**. **RoutedMouseClick**

**Solution** :  
option [3] is correct

**Attempted** :  
option [4] is attempted

##### We are going through a requirement in one of our WPF application where we have to fill  the background of Button control with more than one colour combination.  The brush object I can use for this is \_\_\_\_\_\_.

### **Attempted**

1. LinearGradiantBrush

### **Solution**

1. LinearGradiantBrush

2. SolidBrush

3. ImageBrush

4. GeometryObject

Which Statement is true about RoutedEvent?

**1)**. Routed Event stores only the state of the control

**2)**. Routed Event Stores only the Data when the event is fired

**3)**. **Routed Event stores the state and Data when the event is fired**

**4)**. **All the above**

**Solution** :  
option [4] is correct

**Attempted** :  
option [3] is attempted