

MCQ's 20 thay approximatly sary 12 lecturs tak thay Long Questions yah hay

2 Marks )What are the types of Row & Column Definition?

2 Marks ) Why one should care about Logical Tree, Explain with Example?

**Logical tree exists even if there is no xaml. Properties, events, resources are tied to logical trees. Properties propagated down and events can be routed up or down the tree. Its a simplification of whats actually going on when rendered. Visual tree can be thought of as an extension of the logical tree though some things can be dropped as well. Visual tree exposes visual implementation details e.g. A listbox is a border, two scrollbars and more.**

3 Marks )Why C# let us Develop devices resulation independent graphics?

**Now a days, various type of devices with varying screen sizes are available. C# enable us to develop device independent graphics so that same application/ GUI have same aspect ratio and appearance on all kind of displays. e.g. Windows phone, Windows PC etc.**

**The length converter type converter associated with various length properties supports specifying explicit units of cm, pt, in, or px (the default). Default are device independent pixels. These logical pixels are 1/96 in regardless of screen DPI setting. They are always stored as "double". A typical display is 96 DPI.**

**Important to note that all measurements are DPI independent.**

3 Marks )Name the properties supported by the ScaleTransform also write its default values of each of Them?

### **ScaleTransform**

ScaleTransform enlarges or shrinks an element horizontally, vertically, or in both directions. This transform has four straightforward double properties:

- . **ScaleX**—Multiplier for the element's width (default value = 1)
- . **ScaleY**—Multiplier for the element's height (default value = 1)
- . **CenterX**—Origin for horizontal scaling (default value = 0)
- . **CenterY**—Origin for vertical scaling (default value = 0)

```
<Button.RenderTransform>  
  <ScaleTransform ScaleX="2" ScaleY="2"/>
```

```
</Button.RenderTransform>
```

5 Marks ) Write the code for making 2 buttons with colors 1st is Red and Other is Green?

```
<Window x:Class="midPrep.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="MainWindow" Height="350" Width="525">
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition/>
      <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <Button Grid.Row="0" Name="button1" Background="Red" Foreground="Black" >
      <Label Name="b1">Button 1</Label>
    </Button>
    <Button Grid.Row="1" Grid.Column="1" Name="button2" Background="Green"
  Foreground="Black">
      <Label Name="b2" >Button 2</Label>
    </Button>
  </Grid>
</Window>
```



5 Marks ) C# code is given below write its equivalent XAML Code?

## 5 marks) Name five builtin transform with default values

### RotateTransform

RotateTransform, demonstrated in the preceding section, rotates an element according to the values of three double properties:

- . **Angle**—Angle of rotation, specified in degrees (default value = 0)
- . **CenterX**—Horizontal center of rotation (default value = 0)
- . **CenterY**—Vertical center of rotation (default value = 0)

### ScaleTransform

ScaleTransform enlarges or shrinks an element horizontally, vertically, or in both directions. This transform has four straightforward double properties:

- . **ScaleX**—Multiplier for the element's width (default value = 1)
- . **ScaleY**—Multiplier for the element's height (default value = 1)
- . **CenterX**—Origin for horizontal scaling (default value = 0)
- . **CenterY**—Origin for vertical scaling (default value = 0)

### SkewTransform

SkewTransform slants an element according to the values of four double properties:

- . **AngleX**—Amount of horizontal skew (default value = 0)
- . **AngleY**—Amount of vertical skew (default value = 0)
- . **CenterX**—Origin for horizontal skew (default value = 0)
- . **CenterY**—Origin for vertical skew (default value = 0)

### TranslateTransform

TranslateTransform simply moves an element according to two double properties:

- . **X**—Amount to move horizontally (default value = 0)
- . **Y**—Amount to move vertically (default value = 0)

### MatrixTransform

MatrixTransform is a low-level mechanism that can be used to create custom 2D transforms.

MatrixTransform has a single Matrix property (of type

System.Windows.Media.Matrix) representing a 3x3 affine transformation matrix. In case you're not a linear algebra buff, this basically means that all the previous transforms (or any combination of them) can also be expressed using MatrixTransform.

The 3x3 matrix has the following values:

```
M11 M12 0
M21 M22 0
OffsetX OffsetY 1
```

The final column's values are fixed, but the other six values can be set as properties of the Matrix type (with the same names as shown) or via a constructor that accepts the six values in row-major order.

3 marks) Explain Grid splitter

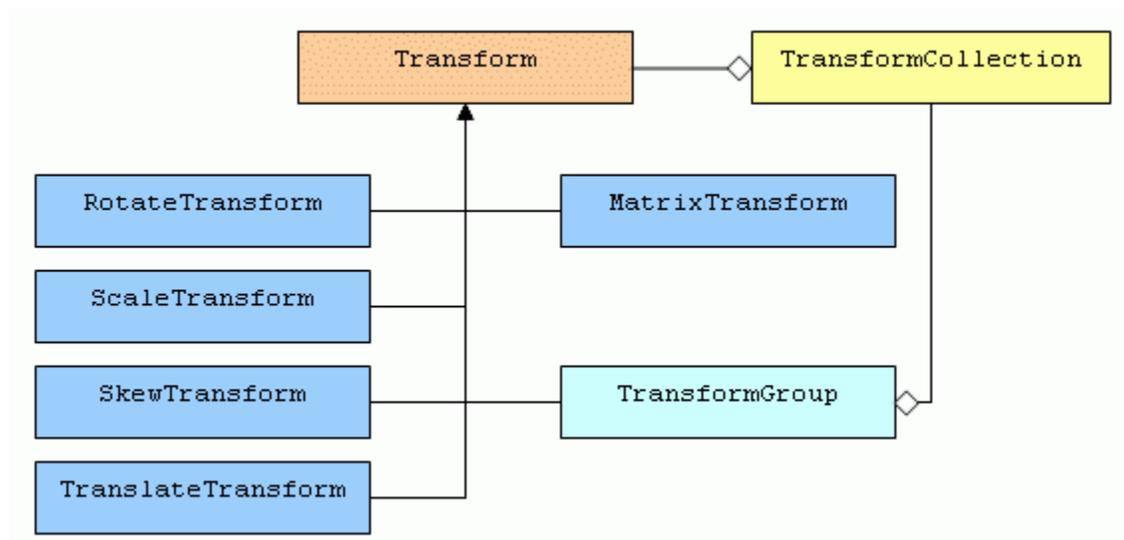
Another attractive feature of Grid is its support for interactive resizing of rows and columns using a mouse or keyboard (or stylus or finger, depending on your hardware). This is accomplished with the GridSplitter class from the same namespace. You can add any number of GridSplitter children to a Grid and give them Grid.Row, Grid.Column, Grid.RowSpan, and/or Grid.ColumnSpan attached property values like any other children. Dragging a GridSplitter resizes at least one cell. Whether the other cells resize or simply move depends on whether they use proportional or nonproportional sizing

For which class WPF are derived? 2

```
using System.Windows;
```

What is the purpose of WPF 2D transform classes? 2

Transformations are available through the System.Windows.Media namespace. The base class for the transformations hierarchy is Transform.



WPF combines the best attribute of some systems. write the names of three. 3

**WPF aims to combine the best attributes of systems**

**DirectX**  
**Windows Forms**  
**Adobe Flash**  
**HTML**

(3D and hardware acceleration),  
(developer productivity),  
(powerful animation support) and  
(declarative markup)

Q. Name 5 built in 2D transform which are supported by "System.Windows.Media" namespace. 5

Built-in transforms are:-

1. Rotate Transform,
2. Scale Transform,
3. Skew Transform,
4. Translate Transform
5. Matrix Transform

Consider the following code snippet and write the output:

```
XmlDocument xmlDoc = new XmlDocument();  
  
xmlDoc.LoadXml("<uni name='Virtual'> university Node</uni.");  
  
console.WriteLine(xmlDoc.DocumentElement.Name);  
  
console.WriteLine(xmlDoc.DocumentElement.InnerText);  
  
console.ReadLine();
```

## How many children an object can have, name them

An object element can have three types of children:

1. A value for a content property,
2. Collection items
3. A Value that can be type-converted to the object element.

## what is the purpose of using xaml extensions?

Markup extensions, like type converters, enable you to extend the expressiveness of XAML. Both can evaluate a string attribute value at runtime (except for a few built-in markup extensions that are currently evaluated at compile time for performance reasons) and produce an appropriate object based on the string. As with type converters, WPF ships with several markup extensions built in.

## aik ZIndex ka question tha,

ZIndex is an integer with a default value of 0 that you can set to any number (positive or negative). Elements with larger ZIndex values are rendered on top of elements with smaller ZIndex values, so the element with the smallest value is in the back, and the element with the largest value is in the front.

aik code tha us me 'ClickMe' name ka method bana k us ko register karna tha or phir us ko call karna tha,

using 'markup extensions' create a button with rectangle and use coding to reset background to null, is tarah aik question tha...

```
<Button xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Background="{x:Null}"
Height="{x:Static SystemParameters.IconHeight}"
Content="{Binding Path=Height, RelativeSource={RelativeSource Self}}"/>
```

Page 33 WPF

**Question: 1 (2 Marks)**

### What is the biggest feature of dependency property?

Arguably the biggest feature of a dependency property is its built-in ability to provide change notification.

### Question: 2 (2 Marks)

When two or more independent grids need to be synchronized, then how can we synchronize them?

**SharedSizeGroup** is then used to keep these three independent Grids in sync when they need to be (that is, the docked case). Figure

### Question: 3 (3 Marks)

Write two differences between WPF and DirectX?

They are two different things. WPF is made for designers to create vector-based applications using XML and managed code. It uses DirectX as the rendering engine under the covers.

DirectX is a set of APIs that abstracts access to modern graphics rendering hardware, as well as other technologies such as sound or input devices.

WPF is not designed for any significant game programming, other than, say, a simple card game. Nor is it designed with the intention of allowing you to create a 3D user interface into your DirectX game.

### Question: 5 (5 Marks)

Write the XAML code to create a “DockPanel” having four buttons with dock settings i.e. Top, Bottom, Right and Left.

```
<DockPanel>
<Button DockPanel.Dock="Top" Background="Red">1 (Top)</Button>
<Button DockPanel.Dock="Left" Background="Orange">2 (Left)</Button>
<Button DockPanel.Dock="Right" Background="Yellow">3 (Right)</Button>
122 CHAPTER 5 Layout with Panels
<Button DockPanel.Dock="Bottom" Background="Lime">4 (Bottom)</Button>
<Button Background="Aqua">5</Button>
</DockPanel>
```

**Question: 6 (5 Marks)**

**What will be the easiest way if someone wants to know whether a certain string is a number or not? Elaborate your answer with an example.**

`IsNumeric(String s)`

This method is used to find if the string is numeric or not.

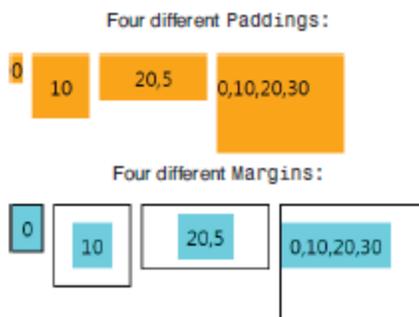
**Difference b/w Margin and padding??? 5 marks ka tha.**

**Margin and Padding**

Margin and Padding are two very similar properties that are also related to an element's size. All FrameworkElements have a Margin property, and all Controls (plus Border) have a Padding property. Their only difference is that Margin controls how much extra space gets placed around the *outside* edges of the element, whereas Padding controls how much extra space gets placed around the *inside* edges of the element.

Both Margin and Padding are of type System.Windows.Thickness, an interesting class that can represent one, two, or four double values. The meaning of these values is demonstrated in Listing 4.1, which applies various Padding and Margin settings to Label controls. The second set of Labels is wrapped in Borders because the margin settings would not be noticeable otherwise. Figure 4.2 shows the rendered result for each Label if each one is individually placed in a Canvas (a panel covered in the next chapter).

Although not shown in this figure, Margin permits negative values. Padding does not.



**1 marks ki coding thi related to triggers.**

Transform ki wo type bato jo sab frame work elements may hoti ha....3 marks ka tha

**Transforms are from the base class System.Windows.Media.Transform. All framework elements have**

- 1. LayoutTransform and**
- 2. RenderTransform**

**Uielements have renderTransformOrigin**

Dock panel may fill k option nahi hoty tu ager hum ko fill karna ha tu kia option use kary gay2 marks ka tha

**We will set Background = "Red"**

These were Mcqs, Its with out choices. But i think enough for idea.

- Every event\_\_\_\_\_ is represented by an event object
- 'Situation' is an event occurrence that requires a(n) reaction\_\_\_\_\_
- Advantages of 'Event Handling' are \_\_\_\_\_ and \_\_\_\_\_
- Console is a/an static\_\_\_\_\_ class in the 'System' namespace
- The range of "sbyte" is \_\_\_\_\_
- When any variable is declared as \_\_\_\_\_ then it is treated as class variable.
- Due to metadata\_\_\_\_\_, "Assembly" Files are self-describing.
- There is/are \_\_\_\_\_ type/s of parameter/s for attribute
- XML document is readable by Machine and human both
- File.Move\_\_\_\_\_ Method is used to move the file from one directory to another.
- For which of the following, WPF does not provide any type converter?
- Compiling XAML involves 3 \_\_\_\_\_ steps
- One of the key goals of WPF and XAML is the separation of front-end and back-end\_\_\_\_\_
- The only xaml\_\_\_\_\_ .Net language had an intrinsic understanding of dependency.
- Routed events are like dependency properties \_\_\_\_\_ on top of .Net properties.

- By convention “tunneling event” names are prefixed with Preview\_\_\_\_\_ and just come before\_\_\_\_\_ the bubbling event occurs.
- We are checking whether an element is eligible for focus or not, which property will help us in this testing?
- Stylus can behave like a mouse\_\_\_\_\_ but has higher\_\_\_\_\_ resolution