



UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 3, Examination in Engineering, August 2015

Module Number: EE3204 Module Name: GUI Programming

Part - II

[1 hour and 45 minutes]

[Answer all questions]

Q1. a) Figure Q1.a, shows a layout in WPF. Explain how to create this using XAML.

[2 marks]

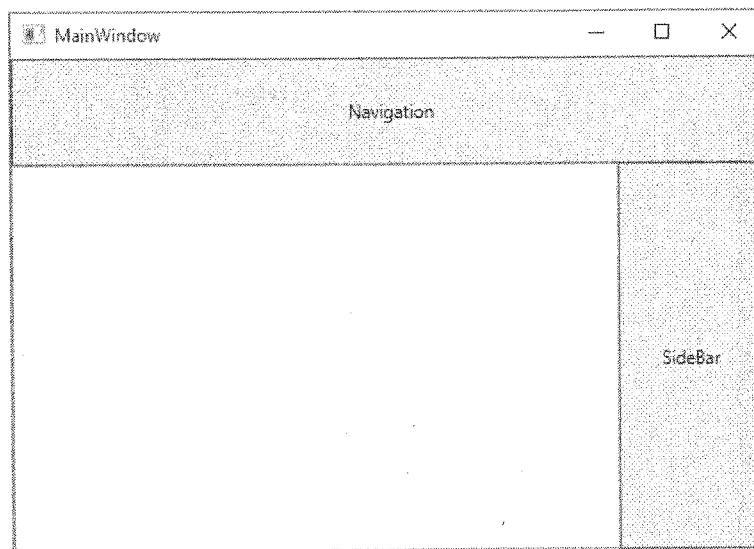


Figure Q1.a: Simple layout example with custom controls

b) Figure Q1.b, shows a layout in WPF. Explain how to create this using XAML.

[2 marks]

c) The Listings 1 shows XAML example code snippets. Draw the output of the code and explain the behavior of it.

[2 marks]

Listing 1: WPF Data Binding example

```
<Window x:Class="DataBinding.MultipleBindings"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="MultipleBindings" Height="300" Width="300"
```

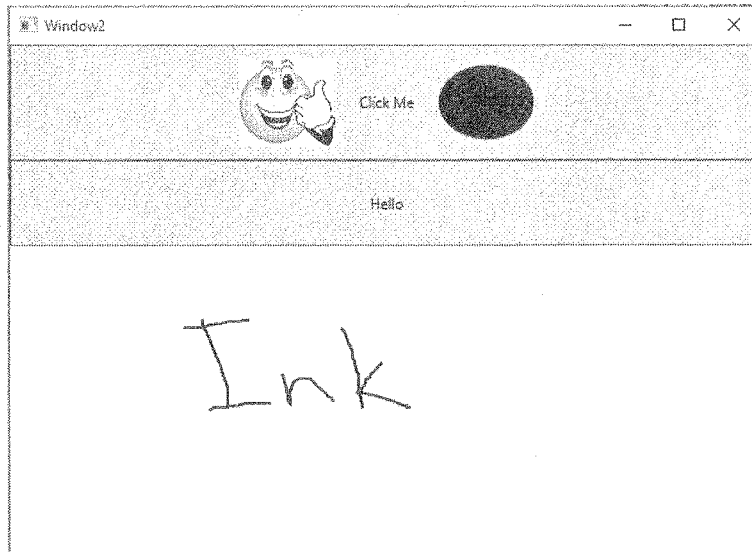


Figure Q1.b: Simple layout example with custom controls

>

```

<Grid Margin="5">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"></RowDefinition>
    <RowDefinition Height="Auto"></RowDefinition>
    <RowDefinition Height="Auto"></RowDefinition>
    <RowDefinition Height="Auto"></RowDefinition>
    <RowDefinition></RowDefinition>
  </Grid.RowDefinitions>

  <Slider Name="sliderFontSize" Margin="3" Minimum="1"
    Maximum="40" Value="10">
  </Slider>

  <TextBox Name="txtContent" Margin="3" Grid.Row="2">Sample
    Content</TextBox>

  <ListBox Margin="3" Grid.Row="3" Name="lstColors">
    <ListBoxItem Tag="Blue">Blue</ListBoxItem>
    <ListBoxItem Tag="DarkBlue">Dark Blue</ListBoxItem>
    <ListBoxItem Tag="LightBlue">Light Blue</ListBoxItem>
  </ListBox>

  <TextBlock Margin="3" Name="lblSampleText"
    FontSize="{Binding ElementName=sliderFontSize, Path=Value}"

```

```

        Grid.Row="4"
        Text="{Binding ElementName=txtContent, Path=Text}"
        Foreground="{Binding ElementName=lstColors, Path=SelectedItem.
            Tag}"
    >
</TextBlock>

</Grid>
</Window>

```

Q2. a) Explain how the rectangle described in XAML code snippets in Listings 2, rendered on the display.

[2 marks]

Listing 2: XAML rectangle code

```

<Rectangle Width="150" Height="100" Grid.Row="4" Margin="5">
    <Rectangle.Fill>
        <LinearGradientBrush StartPoint="0,0" EndPoint="1,1">
            <GradientStop Color="Yellow" Offset="0.0" />
            <GradientStop Color="Red" Offset="0.25" />
            <GradientStop Color="Blue" Offset="0.75" />
            <GradientStop Color="LimeGreen" Offset="1.0" />
        </LinearGradientBrush>
    </Rectangle.Fill>
</Rectangle>

```

b) Explain the output and behavior of the XAML code snippets in Listings 3.

[2.5 marks]

Listing 3: WPF Canvas example

```

<Canvas>
    <Button Padding="10" Name="cmdGrow" Height="40" Width="160"
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Background="Red"
    >

        <Button.Triggers>
            <EventTrigger RoutedEvent="Button.Click">
                <EventTrigger.Actions>
                    <BeginStoryboard>
                        <Storyboard>

```

```

        <DoubleAnimation
            Storyboard.TargetProperty="Width"
            To="300"
            Duration="0:0:5"
            AutoReverse="True"
        />
        <DoubleAnimation
            Storyboard.TargetProperty="Opacity"
            From="1"
            To="0.5"
            Duration="0:0:10"
        />
    </Storyboard>
</BeginStoryboard>
</EventTrigger.Actions>
</EventTrigger>

<EventTrigger RoutedEvent="Button.MouseEnter">
    <EventTrigger.Actions>
        <BeginStoryboard>
            <Storyboard>
                <DoubleAnimation
                    Storyboard.TargetProperty="Height"
                    To="300"
                    Duration="0:0:5"
                    AutoReverse="True"
                />
            </Storyboard>
        </BeginStoryboard>
    </EventTrigger.Actions>
</EventTrigger>

</Button.Triggers>

<Button.Content>
Click Me
</Button.Content>
</Button>
</Canvas>

```

c) Explain what each line of code snippets shown in Listing 4.

[2 marks]

Listing 4: WPF Button click example

```

private void btn_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.InitialDirectory = @"C:\\";
    openFileDialog.Filter = "Text files (*.txt)|*.txt|HTML files (*.html)|*.html";

    if (openFileDialog.ShowDialog() == true)
        txtEditor.Text = File.ReadAllText(openFileDialog.FileName);
}

```

Q3. a) Draw how the output of HTML/CSS code snippets in Listings 5 render the page in the browser.

[2.5 mark]

Listing 5: HTML/CSS page example

```

<!DOCTYPE html>
<html>
  <head lang="en">
    <meta charset="UTF-8">
    <title></title>
    <style>
      div { padding: 2em; }
      #a { border: 3px dashed black; }
      #b { background-color: yellow; }
      .c { float: right; }
      .d { font-weight: bold; }
      .b { float: left; }
      .c, .b {
        border: 5px dotted red;
        margin: 1em; width: 5em;
      }
      p em { text-decoration: underline; }
    </style>
  </head>

  <body>
    <div id="b">
      <p>Hello</p>
      <p>Goodbye</p>
    </div>
  </body>
</html>

```

```

</div>
<div id="a">
  <em>Happy</em> <em>Sad</em>
</div>
<div class="a c"> Monday Tuesday Wednesday</div>
<div class="b"> Thursday Friday Saturday</div>
<div>
<div id="d" class="c">
  <p><em>SUNDAY</em></p>
</div>
</div>
</body>
</html>

```

- b) i) Create a table to show student id, first name, last name and date of birth. Table should show data of five students.
 ii) Draw a border around each cell using CSS.
 iii) Add padding of 5 pixels from top and bottom.
 iv) Make alternating row's background color gray.
- [2 mark]
- c) Show how the code snippets in Listings 6 which use bootstrap css framework renders on a browser on devices with different resolutions.

[2 mark]

Listing 6: Bootstrap layout example

```

<div class="container">
  <div class="row">
    <div class="col-xs-12" style="background-color: red"></div>
  </div>
  <div class="row">
    <div class="col-xs-6" style="background-color: blue"></div>
    <div class="col-xs-6" style="background-color: green"></div>
  </div>
  <div class="row">
    <div class="col-xs-9" style="background-color: blue"></div>
    <div class="col-xs-3" style="background-color: green"></div>
  </div>
  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-4"
      style="background-color: pink"></div>
    <div class="col-xs-12 col-sm-6 col-md-4"
      style="background-color: yellow"></div>
    <div class="col-xs-12 col-md-4"

```

```
        style="background-color: yellow"></div>
    </div>
</div>
```

Q4. a) Explain the usage of following code snippets in HTML/JavaScript.

- i) `document.getElementsByClassName("class_name");`
- ii) `document.querySelector("selector");`
- iii) `document.createElement("element_name");`
- iv) `<button onclick="myFunction()">Click me</button>`

[2 marks]

b) Figure Q4.b shows 4 stages of a grade calculation page for howework assignment. Write the JavaScript code to add behavior to the page as described below. The HTML/CSS code of the page is given in Listing 7.

You will compute the percentage of points earned, with an optional bonus. When "Compute!" is clicked, your JavaScript code should use the values in the text boxes to compute the percentage (rounded to the nearest percent). If the "Bonus +5" checkbox is checked, add +5 percent up to a maximum of 100% total.

You should insert the percentage into the page as a new div added to the end (bottom) of the existing page section with the id of resultsarea. If the overall percentage is 60% or more, give your newly created div a CSS class of pass; otherwise give it a class of fail. Each time the user clicks "Compute!", you will insert such a new div; this means that several divs would be there after several clicks of "Compute!".

In the code shown there are 3 assignments, but your code should work for any number of assignments greater than one. When "Clear" is clicked, all text in all of the input text boxes should be erased.

Assume valid input; that is, assume that when "Compute!" is clicked, the user will have already typed valid text into every box that can be interpreted as an integer.

[4 marks]

Listing 7: Grade Calculator HTML/CSS code.

```
<!DOCTYPE html>
<html>
  <head lang="en">
    <meta charset="UTF-8">
    <title></title>
    <style>
      .pass {
        background-color: #cfc;
```

```

        font-weight: bold;
    }
    .fail {
        background-color: #fcc;
        font-style: italic;
    }
</style>
</head>
<body>
    <h1>Grade Calculator</h1>
    <div id="assignments">
        <div class="hw">
            HW1 <input class="earned" type="text" size="2" />
            / <input class="max" type="text" size="2" />
        </div>
        <div class="hw">
            HW2 <input class="earned" type="text" size="2" />
            / <input class="max" type="text" size="2" />
        </div>
        <div class="hw">
            HW3 <input class="earned" type="text" size="2" />
            / <input class="max" type="text" size="2" />
        </div>
        ...
    </div>
    <div>
        <label><input id="curve" type="checkbox" /> Bonus +5 ?</label>
        >
    </div>
    <div id="resultsarea">
        <button id="compute">Compute!</button>
        <button id="clear">Clear</button>

        <!-- your percentages should be inserted here -->

    </div>
</body>
</html>

```


Grade Calculator

HW /
HW /
HW /
 Bonus +5 ?

Figure (i)

Grade Calculator

HW /
HW /
HW /
 Bonus +5 ?

80

Figure (ii)

Grade Calculator

HW /
HW /
HW /
 Bonus +5 ?

80

85

Figure (iii)

Grade Calculator

HW /
HW /
HW /
 Bonus +5 ?

80

85

59

Figure (iv)

Figure Q4.b: Behavior of HTML/JavaScript/CSS page