1 Basic .NET MAUI Structure

1.1 XAML:

Used for defining the UI

MainPage.xaml

```xaml
<ContentPage
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="MyApp.MainPage">

    <StackLayout>
        <!-- UI Elements Here -->
    </StackLayout>

</ContentPage>
```

1.2 Code-Behind:

Where you write logic in C#.

MainPage.xaml.cs

```csharp
public partial class MainPage : ContentPage
{
    public MainPage()
    {
        InitializeComponent();
    }
}
```

2 Key C# Concepts

2.1 Variables

Main.cs

```csharp
int myNumber = 6;
string myText = "Hello, World!";
```

2.2 Naming Conventions

Private Variables

```csharp
// Uses camelCase
private int myNum = 0;
```

Public Variables

```csharp
// Uses PascalCase
public int OurNum = 0;
```

3 Common UI Elements

Button

```xaml
<Button
    Text="Roll Dice"
    Clicked="OnButtonClicked" />
```

Label

```xaml
<Label Text="Welcome to .NET MAUI!" />
```

Entry

```xaml
<Entry Placeholder="Enter text here" />
```

Slider

```xaml
<Slider Minimum="0" Maximum="100" />
```

Switch

```xaml
<Switch IsToggled="false" />
```

Image

```xaml
<Image Source="dotnet_bot.png" />
```

Button From Code-Behind

```csharp
Button button = new Button
{
    Text = "Click to Rotate Text!",
};
```

4 Event Handling

4.1 In XAML:

MainPage.xaml

```xaml
<Button Text="Roll Dice" Clicked="OnRollDiceClicked" />
```

4.2 In Code Behind:

MainPage.xaml.cs

```csharp
private void OnRollDiceClicked(object sender, EventArgs e)
{
    // Logic for rolling dice
}
```

5 Random Number Generation

Generate a random number (e.g., for dice rolling):

MainPage.xaml.cs

```csharp
Random rand = new Random();
int diceRoll = rand.Next(1, 7); // Generates a number between 1 and 6
```

---

6 Advanced Section

6.1 REST APIs

Used to access data from the internet

REST API Methods

1. **GET** Retrieve info about an item
2. **POST** Create a new item
3. **PUT** Replace info of an item
4. **DELETE** Delete an entire item

6.2 C# API Calls

Using HttpClient:

MainPage.xaml.cs

```csharp
async void GetApiData()
{
    HttpClient client = new HttpClient();
    HttpResponseMessage response = await
        client.GetAsync("https://example.com/api");

    var data =
    JsonConvert.DeserializeObject<ApiResponse>(response);

    // Display the data in your app
    label.Text = data.Data;
}

internal class ApiResponse
{
    public string Data { get; set; }
}
```