FLUTTER SIMPLE APP

The goal for this lecture is to make a simple flutter app the prints hello world in the command console when the button in the center is pressed. In Flutter the app is made from simple structures called **widgets**. Widgets are the basic building blocks used to construct the user interface of an app ,widgets can be simple, like a piece of text or an icon, or more complex, like an entire screen or a button. Let's delve deeper into some of the key widgets used in the provided flutter codes in the app. Let's go through the explanation of each function used in the provided Flutter code:

1. main():
```
void main() {
  runApp(MyApp());
}
```
- The `**main**` function is the entry point of the Dart program. In this case, it calls the `**runApp**` function, which takes an instance of the `**MyApp**` class as an argument. `runApp` sets up the basic Flutter app and starts the execution.

2. MaterialApp:
```
return MaterialApp(
  home: HomeScreen(),
);
```
- `**MaterialApp**` is a top-level widget that configures the overall visual structure of a Material Design app. It provides features like navigation, theming, and other settings.
- The `**home**` property is set to an instance of the `HomeScreen` widget, specifying the initial screen of the app.

3. Scaffold:
```
return Scaffold(
  appBar: AppBar(
    title: Text('Home Screen'),
  ),
  body: Center(
    // ...
  ),
);
```
- `**Scaffold**` is a container widget that provides a basic structure for a screen in a Flutter app. It typically includes an app bar, a body, and other optional elements like a floating action button.
- The `**appBar**` property is set to an `**AppBar**` widget, defining the app bar at the top of the screen.
- The `**body**` property is set to a `**Center**` widget, which centers its child vertically and horizontally.

4. AppBar:
```
appBar: AppBar(
  title: Text('Home Screen'),
),
```
- `**AppBar**` is a material design widget that represents the app bar, typically positioned at the top of the screen.
- The `**title**` property is set to a `**Text**` widget containing the text 'Home Screen', which becomes the title of the app bar.

5. Center:
```
body: Center(
  child: Column(
    // ...
  ),
),
```
- `**Center**` is a widget that centers its child both vertically and horizontally within the available space.

6. Column:
```
child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    // ...
  ],
),
```
- `**Column**` is a widget that arranges its children in a vertical column.
- The `**mainAxisAlignment**` property is set to `MainAxisAlignment.center`, aligning the children at the center vertically.

7. Text:
```
Text(
  'Welcome to My App!',
  style: TextStyle(fontSize: 24),
),
```
- `**Text**` is a widget that displays a piece of text.
- The `**style**` property is set to a `**TextStyle**` widget, specifying the font size as 24.

8. SizedBox:
```
SizedBox(height: 20),
```
- `**SizedBox**` is a box with a specified width, height, or both.
- In this case, it's used to add vertical spacing of 20 pixels between the text and the button.

9. ElevatedButton:
```
ElevatedButton(
  onPressed: () {
    print('Hello, World!');
  },
  child: Text('Say Hello'),
),
```
- `**ElevatedButton**` is a material design button that responds to touches.
- The `**onPressed**` property is set to an anonymous function that prints **'Hello, World!'** to the console when the button is pressed.
- The `**child**` property is set to a `**Text**` widget with the text 'Say Hello', which is the label of the button.

10. MyApp class (StatelessWidget):
```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: HomeScreen(),
    );
  }
}
```
- `**MyApp**` is a stateless widget. It extends the `StatelessWidget` class, indicating that it won't change over time.
- The `**build**` method returns a `**MaterialApp**` widget. `MaterialApp` is a top-level container for a Material Design application. It sets up the basic structure of the app and defines the home screen as an instance of `HomeScreen`.
- The `**@override**` annotation ensures that the `build` method in `MyApp` is intended to override the same method in the `StatelessWidget` class.Using `@override` is not strictly required by the Dart language, but it's considered good practice. It helps in maintaining code clarity and makes it explicit when a method is intended to override a method in a superclass. If there's a mistake, such as a typo in

the method name or a mismatched parameter, the Dart analyzer will catch it at compile-time, reducing the chances of runtime errors related to method overrides.

11. HomeScreen class (StatelessWidget):

```
class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Home Screen'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Welcome to My App!',
              style: TextStyle(fontSize: 24),
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                print('Hello, World!');
              },
              child: Text('Say Hello'),
            ),
          ],
        ),
      ),
    );
  }
}
```

- `**HomeScreen**` is another stateless widget. It represents the content of the home screen.
- The `**build**` method returns a `Scaffold` widget. `**Scaffold**` is a basic structure for a screen in a Flutter app, providing an app bar and a body.
- The `appBar` property is an `AppBar` widget, defining the app bar with the title 'Home Screen'.
- The `body` property contains a `Center` widget, which centers its child vertically and horizontally. Inside the `Center`, there's a `Column` widget with a `Text` widget displaying a welcome message, a `SizedBox` for spacing, and an `ElevatedButton`. The button has an `onPressed` callback that prints 'Hello, World!' to the console when pressed.

12. print('Hello, World!'):
  ```
  onPressed: () {
    print('Hello, World!');
  },
  ```
  - This is an anonymous function assigned to the `onPressed` property of the `ElevatedButton`.
  - When the button is pressed, this function is executed, and it prints 'Hello, World!' to the console.

These are the main functions used in the provided Flutter code, each contributing to the overall structure and behavior of the app.

By: Eng Kondwani Nyirenda
Github User: kondwani0099
Contact: +260960332980
GDG Kitwe