Mobile Application Development
EM6330

KINGSTON UNIVERSITY LONDON in collaboration with ESOFT METRO CAMPUS, Sri Lanka

Deshani Ranasingha deshani28@gmail.com 27 Sep 2024

Mobile Application Development

MODULE CODE: EM6330
LEVEL: 6
CREDITS: 30

Deshani Ranasingha deshani28@gmail.com

27 Sep 2024

Kingston University London

ESOFT
Shaping Lives, Creating Futures.

Introduction to Mobile Application Development using industry standard - Flutter

deshani28@gmail.com

27 Sep 2024

Kingston University London

ESOFT Shaping Lives, Creating Futures.

Dart ??

- Free and open-source.
- Object-oriented programming language.
- Used to develop android, iOS, web, and desktop apps fast.
- Can compile to either native code or javascript.
- Offers modern programming features like null safety and asynchronous programming.
- You can even use Dart for servers and backend.

Language Tour – https://dart.dev/guides/language/language-tour
Language samples - https://dart.dev/samples
Dart cheat sheet lab - https://dart.dev/codelabs/dart-cheatsheet

Flutter ??

Kingston University London

Flutter - It is Google's mobile app **SDK** that build native **iOS, Android, Web and Desktop** app from a **single code base**(One programming language).

ESOFT
Shaping Lives, Creating Futures.

Difference Between Dart & Flutter

Kingston University London

Dart is a client optimized, object-oriented programming language. It is popular nowadays because of flutter. It is difficult to build complete apps only using Dart because you have to manage many things yourself.

Flutter is a framework that uses dart programming language. With the help of flutter, you can build apps for android, iOS, web, desktop, etc. The framework contains ready-made tools to make apps faster.

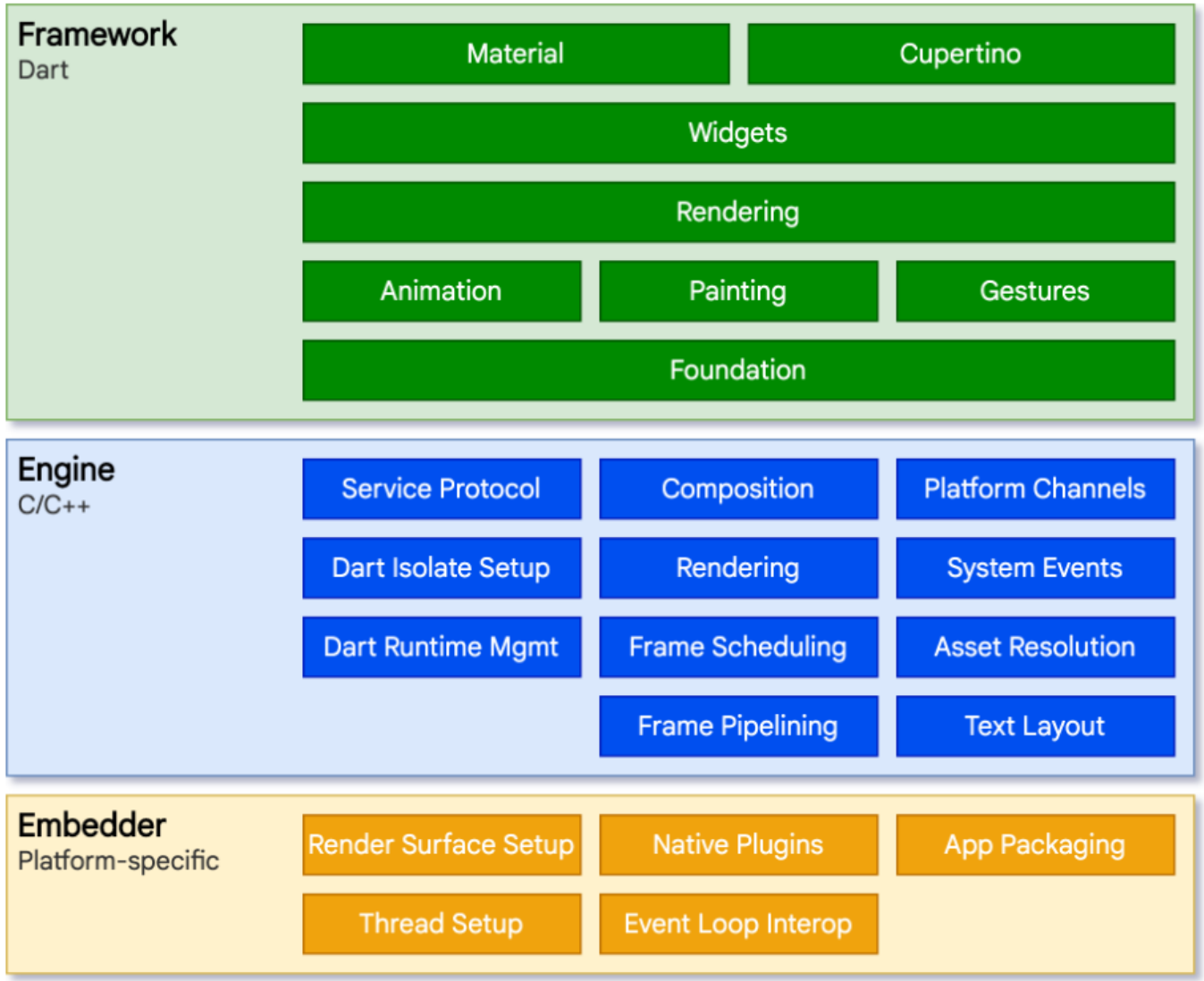
Features of Flutter

- Open-Source
- Cross-platform
- Hot Reload
- Accessible Native Features and SDKs
- Minimal code
- Widgets (Material and Cupertino Design)

Flutter Architecture

- Flutter architecture mainly comprises of fore components.
  1. Flutter Engine
  2. Foundation Library
  3. Widgets
  4. Design Specific Widgets

Flutter Architecture



1. Flutter Engine

- Portable runtime for high-quality mobile apps and primarily based on the C++ language.
- It implements Flutter core libraries that include animation and graphics, file and network I/O, plugin architecture, accessibility support, and a dart runtime for developing, compiling, and running Flutter applications.
- It takes Google's open-source graphics library, **Skia**, to render low-level graphics.

2. Foundation Library

- It contains all the required packages for the basic building blocks of writing a Flutter application.
- These libraries are written in Dart language.

3. Widgets

- In Flutter, everything is a widget, which is the core concept of this framework.
- Widget in the Flutter is basically a user interface component that affects and controls the view and interface of the app.

4. Design Specific Widgets

Kingston University London

- The Flutter framework has two sets of widgets that conform to specific design languages.
- These are **Material Design** for Android application and **Cupertino Style** for IOS application.

Widgets

- The UI of a Flutter app is built out of widgets
- Describe what the view is like given its
- When the state is changed the widget rebuilds its description
- The framework then looks at the difference between new and old to get the minimal list of differences to change to go from one state to the other
- Widgets are classes used to build UI
- Used for both layout and UI elements
- Complex widgets are built from simple widgets
- Example: Text, Row, Column, Container, ListView, AppBar, BottomNavigationBar

Flutter Widget: https://docs.flutter.dev/ui/widgets

Flutter – Stateless Widget

Kingston University London

- Stateless widgets do not require mutable state. it is **immutable**.
- In simple words, Stateless widgets cannot change their state during the runtime of the app, which means the widgets cannot be redrawn while the app is in action.
- The structure of a Stateless widget looks like this:

ESOFT
Shaping Lives, Creating Futures.

Flutter – Stateless Widget

Kingston University London

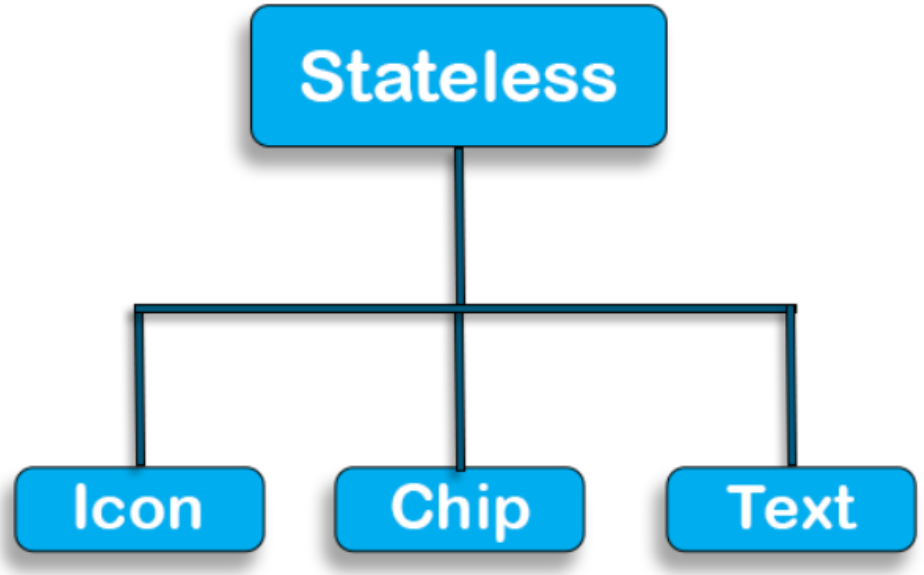
```dart
import 'package:flutter/material.dart';

class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

ESOFT Shaping Lives, Creating Futures.

Flutter – Stateless Widget



Flutter – Stateful Widget

- Stateful widgets have a mutable state, i.e., they are **mutable** and can be drawn multiple times within its lifetime.
- They are the widgets which can change their state multiple times and can be redrawn on to the screen any number of times while the app is in action.
- The structure of a Stateful widget looks like this:

Flutter – Stateful Widget

Kingston University London

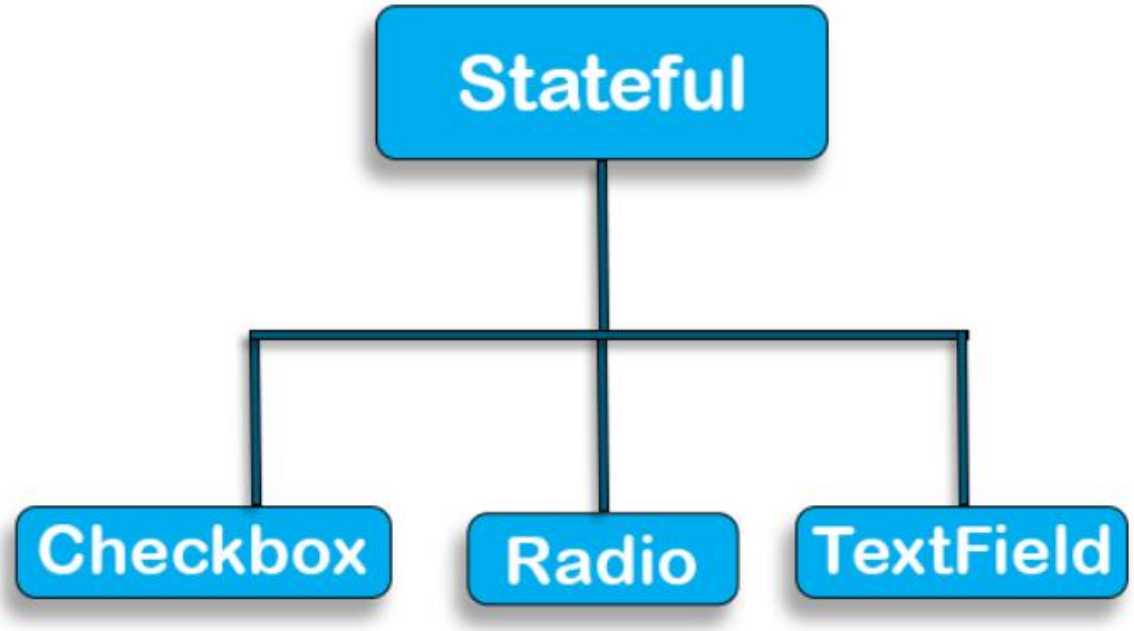
```dart
class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  _HomeScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

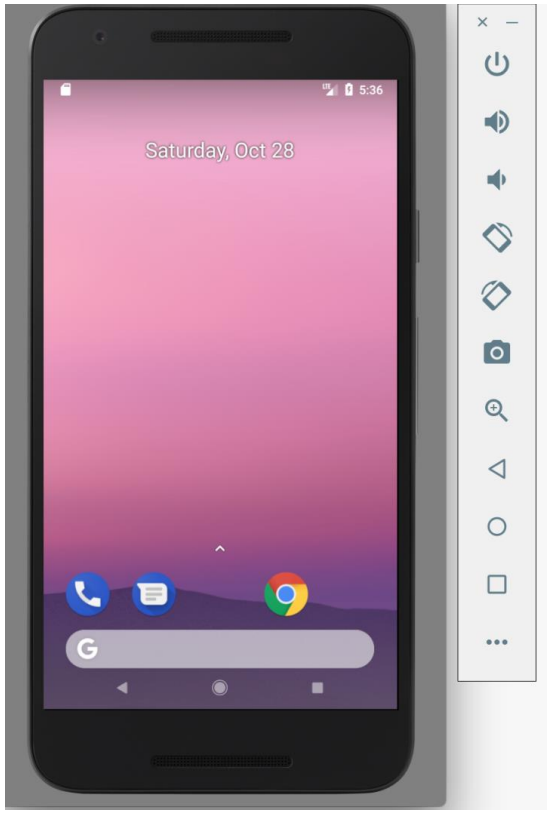
ESOFT
Shaping Lives, Creating Futures.

Flutter – Stateful Widget



The Emulator

- The **Android emulator** is an **Android Virtual Device (AVD),** which represents a specific Android device**.**



Devices

Dart - Basic Dart Program

```
1 void main() {
2     print("Hello World!");
3 }
```

- void **main()** is the starting point where the execution of your program begins.
- Every program starts with a **main** function.
- The curly braces **{}** represent the beginning and the ending of a block of code.
- print**(“Hello World!”);** prints Hello World! on screen.
- Each code statement must end with a semicolon.

Dart - Basic Dart Program

```dart
void main(){
  var firstName = "John";
  var lastName = "Doe";
  print("Full name is $firstName $lastName");
}
```

Dart - Basic Dart Program

```dart
void main() {
int num1 = 10;
int num2 = 3;


int sum = num1 + num2;
int diff = num1 - num2;
int mul = num1 * num2;
double div = num1 / num2;

print("The sum is $sum");
print("The diff is $diff");
print("The mul is $mul");
print("The div is $div");
}
```

Dart - Variable in Dart

String	text values	“John”
Int	integer value	10, -10, 8555 (decimal is not include)
Double	floating point number	10.0, -10.50, 67.90 (decimal is included)
Num	any type of number	10, 20.2, -20 (both in int and double)
Bool	True and False	True and False values
Var	Any values	Bimal’, 12, ‘D’, true
Maps	Ordered group of items	
Lists	represents a set of values as key-value pairs	

Dart - Variable in Dart

Kingston University London

```dart
void main() {
String name = "John";
String address = "USA";
num age = 20;
num height = 5.9;
bool isMarried = false;

print("Name is $name");
print("Address is $address");
print("Age is $age");
print("Height is $height");
print("Married Status is $isMarried");
}

```

ESOFT
Shaping Lives, Creating Futures.

Dart - List

```dart
void main() {
List<String> names = ["Raj", "John", "Max"];
print("Value of names is $names");
print("Value of names[0] is ${names[0]}"); // index 0
print("Value of names[1] is ${names[1]}"); // index 1
print("Value of names[2] is ${names[2]}"); // index 2

int length = names.length;
print("The Length of names is $length");
}
```

Dart - Map

```dart
void main() {
Map<String, String> myDetails = {
    'name': 'John Doe',
    'address': 'USA',
    'fathername': 'Soe Doe'
};

print(myDetails['name']);
}

```

Conditions

Types Of Condition

- **If Condition**
- **If-Else Condition**
- **If-Else-If Condition**
- **Switch case**

If Conditions

```
1 void main(){
2     var age = 20;
3
4     if(age >= 18){
5       print("You are voter.");
6     }
7 }
```

If-Else Conditions

```
1 void main(){
2   int age = 12;
3   if(age >= 18){
4     print("You are voter.");
5   }else{
6     print("You are not voter.");
7   }
8 }
```

If-Else-If Conditions

Kingston University London

```
1  void main() {
2    int noOfMonth = 5;
3
4    if (noOfMonth == 1) {
5      print("The month is jan");
6    } else if (noOfMonth == 2) {
7      print("The month is feb");
8    } else if (noOfMonth == 3) {
9      print("The month is march");
10   } else if (noOfMonth == 4) {
11     print("The month is april");
12   } else if (noOfMonth == 5) {
13     print("The month is may");
14   } else if (noOfMonth == 6) {
15     print("The month is june");
16   } else if (noOfMonth == 7) {
17     print("The month is july");
18   } else if (noOfMonth == 8) {
19     print("The month is aug");
20   } else if (noOfMonth == 9) {
21     print("The month is sep");
22   } else if (noOfMonth == 10) {
23     print("The month is oct");
24   } else if (noOfMonth == 11) {
25     print("The month is nov");
26   } else if (noOfMonth == 12) {
27     print("The month is dec");
28   } else {
29     print("Invalid option given.");
30   }
31 }
```

ESOFT Shaping Lives, Creating Futures.

If-Else-If Conditions

Kingston University London

```
1  void main() {
2    int noOfMonth = 5;
3
4    if (noOfMonth == 1) {
5      print("The month is jan");
6    } else if (noOfMonth == 2) {
7      print("The month is feb");
8    } else if (noOfMonth == 3) {
9      print("The month is march");
10   } else if (noOfMonth == 4) {
11     print("The month is april");
12   } else if (noOfMonth == 5) {
13     print("The month is may");
14   } else if (noOfMonth == 6) {
15     print("The month is june");
16   } else if (noOfMonth == 7) {
17     print("The month is july");
18   } else if (noOfMonth == 8) {
19     print("The month is aug");
20   } else if (noOfMonth == 9) {
21     print("The month is sep");
22   } else if (noOfMonth == 10) {
23     print("The month is oct");
24   } else if (noOfMonth == 11) {
25     print("The month is nov");
26   } else if (noOfMonth == 12) {
27     print("The month is dec");
28   } else {
29     print("Invalid option given.");
30   }
31 }
```

ESOFT Shaping Lives, Creating Futures.

Conditions

Kingston University London

```
1  void main(){
2    int num1 = 1200;
3    int num2 = 1000;
4    int num3 = 150;
5
6    if(num1 > num2  && num1 > num3){
7      print("Num 1 is greater: i.e $num1");
8    }
9    if(num2 > num1 && num2 > num3){
10     print("Num2 is greater: i.e $num2");
11   }
12   if(num3 > num1 && num3 > num2){
13     print("Num3 is greater: i.e $num3");
14   }
15 }
```

ESOFT
Shaping Lives, Creating Futures.

Switch

```
void main() {
  var dayOfWeek = 5;
  switch (dayOfWeek) {
    case 1:
        print("Day is Sunday.");
        break;
    case 2:
        print("Day is Monday.");
      break;
    case 3:
      print("Day is Tuesday.");
      break;
    case 4:
        print("Day is Wednesday.");
      break;
    case 5:
        print("Day is Thursday.");
      break;
    case 6:
        print("Day is Friday.");
      break;
    case 7:
        print("Day is Saturday.");
      break;
    default:
        print("Invalid Weekday.");
      break;
  }
}
```

Switch

```
void main() {
  var dayOfWeek = 5;
  switch (dayOfWeek) {
    case 1:
        print("Day is Sunday.");
        break;
    case 2:
        print("Day is Monday.");
      break;
    case 3:
      print("Day is Tuesday.");
      break;
    case 4:
        print("Day is Wednesday.");
      break;
    case 5:
        print("Day is Thursday.");
      break;
    case 6:
        print("Day is Friday.");
      break;
    case 7:
        print("Day is Saturday.");
      break;
    default:
        print("Invalid Weekday.");
      break;
  }
}
```

Ternary Operator

The ternary operator is like if-else statement. This is a one-liner replacement for the if-else statement. It is used to write a conditional expression, where based on the result of a boolean condition, one of the two values is selected.

```
condition ? exprIfTrue : exprIfFalse
```

Ternary Operator

```
1 void main() {
2   int num1 = 10;
3   int num2 = 15;
4   int max = 0;
5   if(num1> num2){
6     max = num1;
7   }else {
8     max = num2;
9   }
10  print("The greatest number is $max");
11 }
12
```

```
1 void main() {
2   int num1 = 10;
3   int num2 = 15;
4   int max = (num1 > num2) ? num1 : num2;
5   print("The greatest number is $max");
6 }
```

Dart Loops

There are different types of loop in Dart. They are:

- **For Loop**
- **For Each Loop**
- **While Loop**
- **Do While Loop**

```
1 void main() {
2   print("John Doe");
3   print("John Doe");
4   print("John Doe");
5   print("John Doe");
6   print("John Doe");
7   print("John Doe");
8   print("John Doe");
9   print("John Doe");
10  print("John Doe");
11  print("John Doe");
12  print("John Doe");
13 }
```

For Loop

Kingston University London

```
1 void main() {
2   for (int i = 0; i < 10; i++) {
3     print("John Doe");
4   }
5 }
```

ESOFT
Shaping Lives, Creating Futures.

For Each Loop

Kingston University London

```
 4 void main() {
 5   List<String> footballplayers = ['Ronaldo', 'Messi', 'Neymar', 'Hazard'];
 6   footballplayers.forEach( (names){
 7     print(names);
 8    }
 9   );
10 }
11
```

```
 4 void main(){
 5   List<String> footballplayers=['Ronaldo','Messi','Neymar','Hazard'];
 6   footballplayers.forEach(
 7     (names)=>print(names)
 8   );
 9 }
```

ESOFT Shaping Lives, Creating Futures.

While Loop

```
1 void main() {
2   int i = 1;
3   while (i <= 10) {
4     print(i);
5     i++;
6   }
7 }
```

Do While Loop

```
1 void main() {
2    int i = 1;
3    do {
4       print(i);
5       i++;
6    } while (i <= 10);
7 }
```

Dart Functions

Kingston University London

```
returntype functionName(parameter1,parameter2, ...){
  // function body
}
```

Return type: It tells you the function output type. It can be void, String, int, double, etc. If the function doesn't return anything, you can use void as the return type.

Function Name: You can name functions by almost any name. Always follow a lowerCamelCase naming convention like void printName().

Parameters: Parameters are the input to the function, which you can write inside the bracket (). Always follow a lowerCamelCase naming convention for your function parameter.

Example: 1

```
1 void main(){
2   printName();
3 }
4
5 void printName(){
6   print("My name is Amal. I am from function.");
7 }
```

Example: 2

```
1 void add(int num1, int num2){
2    int sum = num1 + num2;
3     print("The sum is $sum");
4 }
5
6 void main(){
7    add(10, 20);
8 }
```

Q&A

deshani28@gmail.com

27 Sep 2024

Kingston University London

ESOFT Shaping Lives, Creating Futures.

Thank you!

deshani28@gmail.com

27 Spe 2024

Kingston University London

ESOFT Shaping Lives, Creating Futures.