CSF306 - Labs

Lab Tasks - Revision

End Term Topics

# Asynchronous Programming

a)

1

void async Testing () { print ( *' 1 '* );

Future (() => print ( *' 2 '* )). then ( ( value ) => print ( *' 3 '* ),

);

Future . microtask (() => print ( *' 4 '* ));

Future . microtask (() => print ( *' 5 '* )); Future . delayed (

Duration ( seconds : 1) , () => print ( *' 6 '* ),

);

Future (() => print ( *' 7 '* )). then (

( value ) => Future (() => print ( *' 8 '* )),

);

Future (() => print ( *' 9 '* )). then ( ( value ) => Future . microtask (

() => print ( *' 10 '* ),

),

);

print ( *' 11 synchronous '* );

}

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

Read the above code and understand how the Future(), Future.microTask() and Future.delayed() works. Once you have understood what each line does, answer the following question.

* What will be the output if the above code was executed from a main method.
* Does the output order change if you modified line 16 from Future.microtask to a Future call? i.e. Future.microtask is replaced with Future.
* Without having to change any of the Future calls, what changes need to be done to the function to get the output in the ascending order of the numbers, i.e. the output should print 1, 2,3...11.

b)

1

void more Async () { Future <int >. delayed (

Duration ( seconds : 1) ,

() => 42 ,

)

. then (

( value ) => print ( *' Value : $value '* ), on Error : ( e) {

print ( *' on Error : $e '* );

},

)

. catch Error (

( error ) => print ( *' catch Error : $error '* ),

)

. when Complete (

() => print ( *' Future is complete '* ),

);

}

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

As part of this code, we are going to check out the Future object and its various callbacks. Answer the following questions based on the code given above.

* What is the output of moreAsync() function execution. Trace the lines that are being executed.
* Change the line 7 to

(value) => Future.error(‘oh no there is an error’)

Now that we have explicitly thrown an error in our code, which of the two error handling mechanisms (onError, catchError) will be called? Explain why.

* Change the code so that the other error handling mechanism can be called. For this you need to know when each of these (onError and catchError) will be called.

# Unit Testing:

Testing the calculator mobile app.

You are provided with a sample calculator application, which works for simple calculations.

* + Write a dart unit test which tests all the functions provided in the Calculations (not the widget but the class that does the calculator operations). Ensure you test all the functions available as part of the Calculations and group them together.
  + Write a widget test which adds 5 and 5 and checks if the result is set to 10.0 using widget testing.

**Note:** Some of the actions like add, subtract, and equals are represented as icons instead of text or buttons with text. This is to ensure that you can use different finders to do your testing.

The code can be found at [the Github location](https://github.com/rama-vaidhiy/flutter_examples/tree/main/calculatortesting).

# Google Firebase Access

While trying to add a Google Firestore database to a Flutter mobile application, the following files (google- services.json for Android and GoogleService-Info.plist for iOS) were provided.

## google-services.json

{

"project\_info": { "project\_number": "867007033627",

"project\_id": "testfirebase-7c6f4", "storage\_bucket": "testfirebase-7c6f4.appspot.com"

},

"client": [

{

"client\_info": {

"mobilesdk\_app\_id": "1:867007033627:android:952af55eff64ffd712e3be", "android\_client\_info": {

"package\_name": "csf306.revision.labs"

}

},

"oauth\_client": [

{

"client\_id": "867007033627-na1c7g8gav1055dha22o756rh7utbgsr.apps.googleusercontent.com", "client\_type": 3

}

],

"api\_key": [

{

"current\_key": "AIzaSyDFHpbJ-VCPmOnSG5XN0SeHg8IMhwY3msc"

}

],

"services": { "appinvite\_service": {

"other\_platform\_oauth\_client": [

{

"client\_id": "867007033627-na1c7g8gav1055dha22o756rh7utbgsr.apps.googleusercontent.com", "client\_type": 3

}

]

}

}

}

],

"configuration\_version": "1"

}

## GoogleService-Info.plist

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" ["http://www.apple.com/DTDs/PropertyList-1.0.dtd">](http://www.apple.com/DTDs/PropertyList-1.0.dtd)

<plist version="1.0">

<dict>

<key>CLIENT\_ID</key>

<string>867007033627-dc3gr0sd29058r4d98713hn44cgb33ak.apps.googleusercontent.com</string>

<key>REVERSED\_CLIENT\_ID</key>

<string>com.googleusercontent.apps.867007033627-dc3gr0sd29058r4d98713hn44cgb33ak</string>

<key>API\_KEY</key>

<string>AIzaSyBFBTqsmesmivds4aVoT1KCW-bGBSvueAA</string>

<key>GCM\_SENDER\_ID</key>

<string>867007033627</string>

<key>PLIST\_VERSION</key>

<string>1</string>

<key>BUNDLE\_ID</key>

<string>csf306.revision.labs</string>

<key>PROJECT\_ID</key>

<string>testfirebase-7c6f4</string>

<key>STORAGE\_BUCKET</key>

<string>testfirebase-7c6f4.appspot.com</string>

<key>IS\_ADS\_ENABLED</key>

<false></false>

<key>IS\_ANALYTICS\_ENABLED</key>

<false></false>

<key>IS\_APPINVITE\_ENABLED</key>

<true></true>

<key>IS\_GCM\_ENABLED</key>

<true></true>

<key>IS\_SIGNIN\_ENABLED</key>

<true></true>

<key>GOOGLE\_APP\_ID</key>

<string>1:867007033627:ios:a107667f5a8d828212e3be</string>

</dict>

</plist>

Answer the following questions based on the contents of these files.

* What is the purpose of these files with respect to the Flutter mobile app which uses them?
* What should the package name of your Flutter mobile application be called in order to use the Google Firestore database?
* Is it possible to know if there is a sign-in required for the Firestore from these files? If so, what is it?

# Flutter architecture and State Management

Flutter’s [state management documentation](https://docs.flutter.dev/development/data-and-backend/state-mgmt/intro) has the detailed list of articles you need to understand the various ways you can manage state.

The default **Counter** Flutter application has been changed to use the **provider** plugin based state management using the steps described in the above link and the source code is available on the [Github](https://github.com/rama-vaidhiy/flutter_examples/blob/main/withprovider.dart) location with necessary comments in the code.

Go through the document and the code and answer the following questions:

* + The observer design pattern is in use in the given sample code. Identify the Observer and the Observable classes/objects.
  + How is the subscription to the observable object handled?
  + **Extension** If I wanted to get the initial value of the counter from the user (using a dialog for example) and then set the counter value to the user defined value and then perform the increment action, the current code will not work. What changes should we do to make it work. **Note:** this might require a few changes both in the observable class and to the way we subscribe to it and you might have to look at the [provider documentation](https://pub.dev/documentation/provider/latest/provider/provider-library.html) to help you out.

[FlutterSamples.com](https://fluttersamples.com/)—This site is a collection of Flutter apps that demonstrate different ways to approach the architecture of your Flutter app and its state. It includes sample apps that use blocs, Redux, inherited widgets, and more.