5COSC023W - Tutorial 6 Exercises

As part of this tutorial for this week, you should complete **ALL** the tasks described in the following specifications: (**make sure that you ask questions to your tutor for anything that you do not understand or if you are stuck at any point**).

> Tutorial sessions are practical sessions that you need to work towards the exercises set. They will give you the chance to practice the material learned in the lectures and learn new things as well.
>
> **You should not use these sessions to work towards the assessed assignments!** If you decide to work towards your assessed work instead, then you are not considered as part of the tutorial session. You will not get any help on the code of the assessed work by your tutor but you can ask your tutor ONLY about any clarifications you might need regarding the specification of the coursework.

Like all other modules, you are expected to study towards you module outside the lecture and dedicated tutorial slots for a number of hours. If you do not finish all of the exercises in the tutorial session, make sure that you finish them on your own time and by the end of the week. This is a normal process and part of your university learning.

For all the tasks you should use Jetpack Compose and NOT Views!

1 Activities and Intents

Implement an Android application which has 2 activities (i.e. 2 Activity classes). The first activity displays 1 button and a list of selectable strings corresponding to images (e.g. you can use the filenames associated with the dog images that you used in the last tutorial exercises).

When the user selects from the list an image name and presses the button, a second activity starts and replaces the current one. The second activity displays an image corresponding to the filename selected in the first activity and also contains a button which when pressed starts a new instance of the first activity.

2 Communication between Activities

As we have already seen, one activity can start a new one and pass to it certain data by using the `putExtra` methods of the `Intent` class.

Similarly the new activity that will be started can pass certain data (e.g. calculations) back to the calling activity.

The following are the required steps for the bidirectional communication of two activities:

1. Call `rememberLauncherForActivityResult()` from inside a composable to register a launcher for a second activity. The trailing lambda describes what will happen after the launcher starts and the second activity terminates (the lambda is the callback function of the second activity after termination).

2. Call the launcher of the second activity from a listener such as the `onClick()` of a `Button` composable.

3. The second activity does its job (calculations, etc.) and then:

   (a) it sets the result code by using the `setResult(int)` or `setResult(int, Intent)` methods. The result code can have the values `RESULT_OK` or `RESULT_CANCELED` (the first value ia returned if the second activity calculated and returned explicitly that value. The latter if the second activity did not return any result or the second activity crashed).

   The second argument of the second version of the overloaded `setResult` method is an Intent that can be used to pass additional data to the calling activity (first activity). Data to the intent are attached by the second activity using the usual `putExtra` methods.

   (b) it calls the `finish()` method to terminate.

   (c) The calling activity will receive that `Intent` object and can extra data from it. Data to the intent are attached by the second activity using the usual `putExtra` methods we have seen previously.

The following example illustrates all of this.

The application creates a first activity and passes 10 and 35 to the second activity. The second activity calculates the sum of the two numbers and returns the result (sum) by attaching it to the intent used to communicate with the first activity:

```
resultIntent.putExtra("sum", sum)
setResult(RESULT_OK, resultIntent)
```

Make sure that you type all of the code and ask your tutor if you do not understand something in it.

The first activity:

```
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.rememberLauncherForActivityResult
import androidx.activity.compose.setContent
import androidx.activity.result.contract.ActivityResultContracts
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.height
import androidx.compose.material3.Button
```

```
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.unit.dp

class MainActivity : ComponentActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       setContent {
           GUI()
       }
   }

   @Composable
   fun GUI() {
      var sum_result: Int? by remember { mutableStateOf(0) }

      val startForResult =
      rememberLauncherForActivityResult(ActivityResultContracts.StartActivityForResult()) {
               result ->
                if (result.resultCode == RESULT_OK) {
                    sum_result = result.data?.getIntExtra("sum", 0)
                }
      }

      val context = LocalContext.current
      Column(
           Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
      ) {
            Button(onClick = {
                val intent = Intent(context, SecondActivity::class.java)
                intent.putExtra("number1", 10)
                intent.putExtra("number2", 35)
                startForResult.launch(intent)
            }) {
                Text("Start second activity")
            }
            Spacer(modifier=Modifier.height(32.dp))
            Text("Received from second activity: $sum_result")
        }
    }
}
```

The second activity:

```
package uk.ac.westminster.activitiesbidirectionalcommunicationcomposables

import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.sp

class SecondActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        val number1 = intent.getIntExtra("number1", 0)
        val number2 = intent.getIntExtra("number2", 0)
        val sum = number1 + number2

        setContent {
            GUI2(sum)
        }
    }

    @Composable
    fun GUI2(sum: Int) {
        Column (Modifier.fillMaxSize(), verticalArrangement = Arrangement.Center){
            Text(
                "Second Activity, sum is $sum", fontSize = 24.sp
            )
            Button(onClick = {
                val resultIntent = Intent()
                resultIntent.putExtra("sum", sum)
                setResult(RESULT_OK, resultIntent)
                finish()
            }) {
                Text("Finish Activity")
            }
        }
    }
}
```

3 Extending the App "Communication between Activities"

Extend the previous application so that the user can type 2 numbers (using two `Textfield`s) and these will be sent to the second activity.