5COSC023W - Tutorial 4 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**).

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.

1 Creating User text input

Implement an application which contains a textbox asking the user to enter their name. The application displays the messame "Hello name, how are you doing?" where name is the user text input.

Hint: Use the `TextField` composable function (similar to the `Button` composable we have seen in the lecture and you used in the previous exercises).

The arguments accepted in a `TextField` are:

```
TextField(value = name, onValueChange = {})
```

where `value` is what the textbox displays (in the above example it is set to the value of a variable called `name`). The `onValueChange` contains the code that it will be executed every time that the user starts typing something.

- You should create a state for this composable with a variable called `name`.
- Change its value within the `onValueChange` argument of the `TextField`. To do so, assign to `onValueChange` a lambda expression accepting one argument and setting the value of `name` to the single parameter of the lambda expression.

  Hint: If you don't recall what is a lambda expression, look at the lecture material to remind yourself.

2 Extending the Lottery Program

Extend the Lottery application that you developed in the last tutorial, so that it contains a textbox. The user will type some numbers (separated by spaces and/or commas) in the textbox and the program will exclude these numbers from the generated results.

Hint: Use the `split` function of a String. The function splits a string into tokens based on a delimiter passed to it as an argument (e.g. this could be a space or other sequence of chars or even a regular expression). It returns a list of strings with all the tokens

Lookup at the relevant documentation of the `split` function:

`https://kotlinlang.org/api/core/kotlin-stdlib/kotlin.text/split.html`

3 An Employee Tracking Application

1. Implement an Android application which asks the user to enter the details of employees in a company (employee ID, name, address, age, position, salary).

   The application will store these in a list of employee objects in memory. You should create an appropriate Employeee class and use this to create the objects.

2. Extend the application by creating a button which displays all the employees entered in a `Text` composable.

3. Create a second button which displays all the employees in descending order according to the salary.

   Hint: Look at the `sortedBy` and `sortedByDescending` functions at the official Kotlin documentation:

   `https://kotlinlang.org/docs/collection-ordering.html#custom-orders`

4. Modify the employee application so that the details of an employee previously entered in the system can be modified and then updated in the list.

5. Modify the application so that the user will be able to delete individual employees from the list based on their ID.