Week 1 : Session 5

Functions and Nullable Values

Exercise

4
_____

[Click] → Button

→ EditText

1@2@3@4 → textView

TestDemo

7:28

6

CLICK

1@2@3@4@5@6@
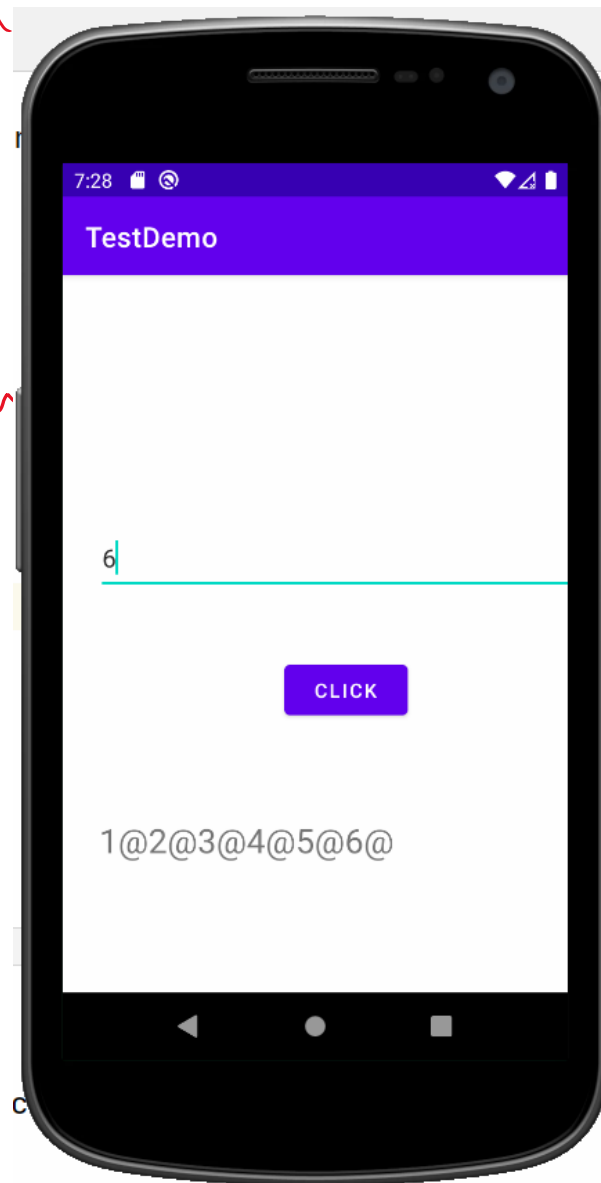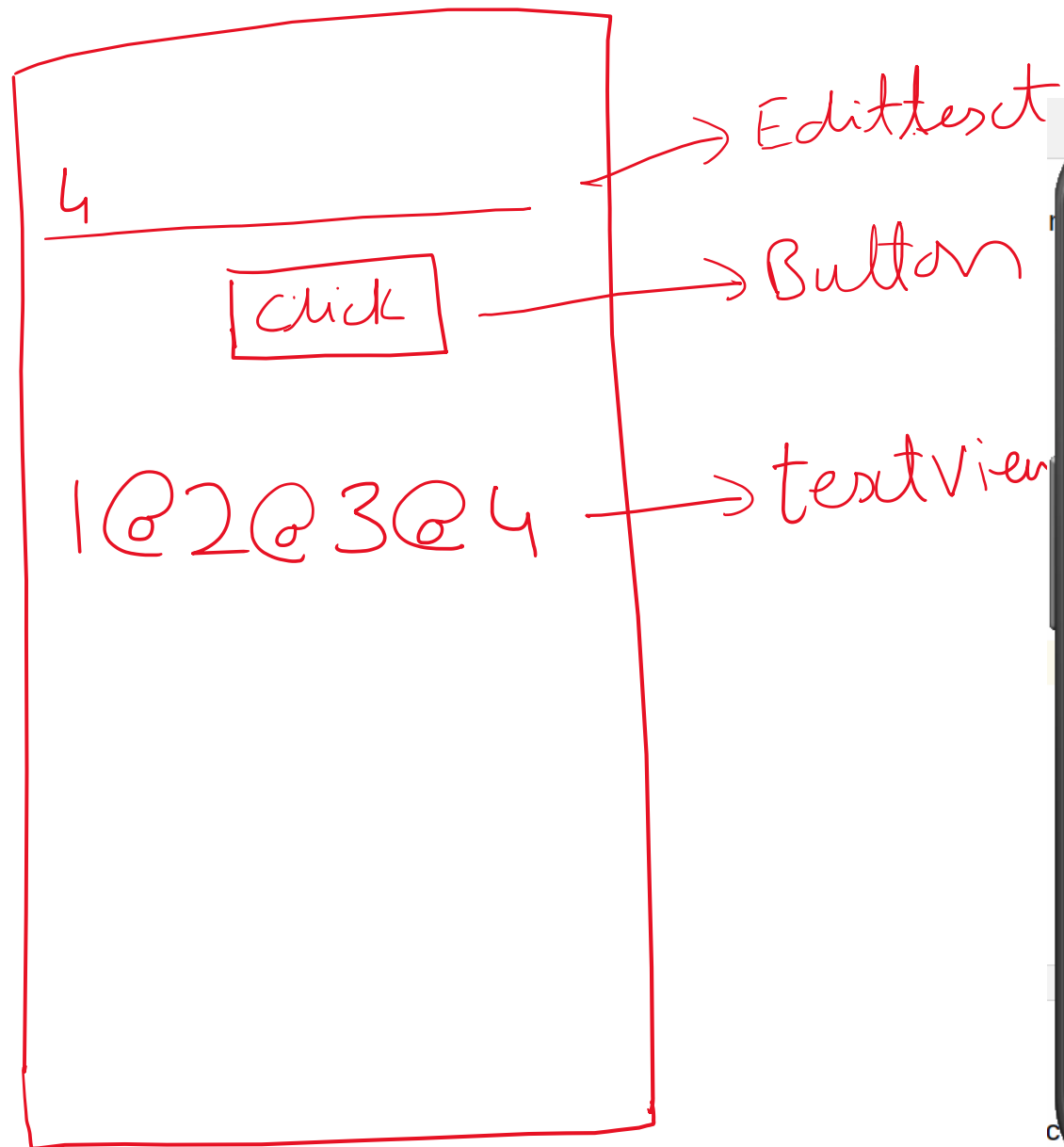
Agenda

- Functions

- Kotlin Null Safety

- Exercise : Demo Application

Functions

- Function is a group of related statements that perform a specific task.

- Functions are used to break a large program into smaller and modular chunks, makes our program more organized and manageable.

- Functions avoids repetition and makes code reusable.

- Type of Functions
  - Kotlin Standard Library Function
    - are built-in functions in Kotlin that are readily available for use
    - print(), main(), sqrt()
  - User-defined functions
    - create functions yourself

```kotlin
3   fun main(){
4       var number = 2
5       print("Result = ${Math.sqrt(number.toDouble())}")
6   }
```

```kotlin
3   fun main(){
4       greet()
5   }
6
7   fun greet(){
8       print("Hello World")
9   }
```

Functions Examples

return type (handwritten)
parameter (handwritten)

```
3      fun addNumbers(n1: Double, n2: Double): Int {
4          val sum = n1 + n2                      ① (handwritten)
5          return sum.toInt()                     ② (handwritten)
6      }
7
8  ▶  fun main() {
9          val number1 = 12.2
10         val number2 = 3.4
11
12         val result: Int = addNumbers(number1, number2)
13         println("result = $result")
14     }
```

arguments (handwritten)

You can omit the curly braces { } of the function body and specify the body after = symbol if the function returns a single expression

```
3  ▶  fun main() {
4          println(getName("John", "Doe"))
5      }
6
7      fun getName(firstName: String, lastName: String): String = "$firstName $lastName"
```

argument (handwritten)

Kotlin Null Safety

- Kotlin null safety is a procedure to eliminate the risk of null reference from the code.

- NULLABLES/OPTIONALS in Kotlin

- Kotlin supports nullability as part of its type System.
  - That means You have the ability to declare whether a variable can hold a null value or not.
  - By supporting nullability in the type system, the compiler can detect possible NullPointerException errors at compile time and reduce the possibility of having them thrown at runtime.

- Handling Kotlin Nullable types
  - Nullable types are declared by putting a ? behind the String

```
3  ▶   fun main(){
4          var str: String? = "Hello" // variable is declared as nullable
5          str = null
6          print(str)
7      }
```

Exercise

8:29

TestDemo

10   +   ▼   2

CALCULATE

Result :

Edit text

10   2

Spinner

+
−
✳
/

Calculate   Button

Result : 12   Textview