TASK GUIDE (B1.04)

A. Objectives.

Students can make fields and methods in main activity class.

B. Requirements.

Hardware:

- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality

Software:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- JDK 8
- Android Studio IDE 3.5

C. Resources.

Documents:

- Guide

Supplement files:

- ViewTest.java

Test code:

- TestB1BasicActivityKT041.java

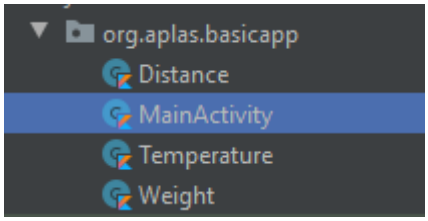
D. Task Description.

Student start to define fields and methods in main activity class.

Reference to learn Kotlin:
https://kotlinlang.org/docs/reference/android-overview.html

E. Guidance.

1. Open BasicApp project that already test passed.
2. Open MainActivity.kt.



3. Make some fields with this description.

Name	Data type	Modifiers access	Contructed
dist	Distance	private	yes
weight	Weight	private	yes
temp	Temperature	private	yes
convertBtn	Button	private	no
inputTxt	EditText	private	no
outputTxt	EditText	private	no
unitOri	Spinner	private	no
unitConv	Spinner	private	no
unitType	RadioGroup	private	no
roundBox	CheckBox	private	no
formBox	CheckBox	private	no
imgView	ImageView	private	no

```
private val dist = Distance()
private val weight = Weight()
private val temp = Temperature()
private var convertBtn: Button? = null
private var inputTxt: EditText? = null
private var outputTxt: EditText? = null
private var unitOri: Spinner? = null
private var unitConv: Spinner? = null
private var unitType: RadioGroup? = null
private var roundBox: CheckBox? = null
private var formBox: CheckBox? = null
private var imgView: ImageView? = null
```

4. Make 2 blank methods with this description.

Name	Return data type	Modifiers access	parameters type
convertUnit	double	protected	String, String, String, double
strResult	String	protected	double, boolean

```
   protected fun convertUnit(){
   }
   protected fun strResult(`val`: Double, rounded: Boolean):
String {
   }
```

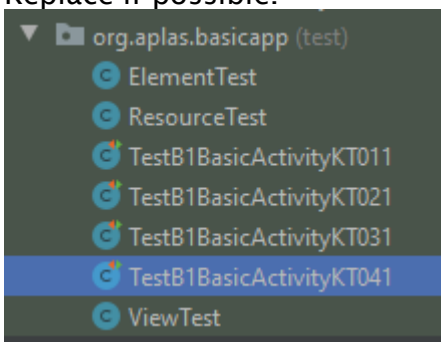
5. "convertUnit" method (function) will return a double value that convert from a unit (original) to another one (converted) as a return value. Utilize the "convert" methods in Temperature, Distance, and Weight class to create algorithm of this method. There are 4 parameters:
    - String type, as original unit (possibility values are *Temperature, Distance, Weight*).
    - String? oriUnit, as original unit (possibility values are *˚C,˚F,K,Mtr,Inc,Mil,Ft, Grm,Pnd,Onc*).
    - String? convUnit, as converted unit (possibility values are *˚C,˚F,K,Mtr,Inc, Mil,Ft,Grm,Pnd,Onc*).
    - double value, as a original value that will converted.

```
protected fun convertUnit(
        type: String,
        oriUnit: String?,
        convUnit: String?,
        value: Double
    ): Double {
        return if (type == "Temperature") {
            temp.convert(oriUnit!!, convUnit!!, value)
        } else if (type == "Distance") {
            dist.convert(oriUnit!!, convUnit!!, value)
        } else {
            weight.convert(oriUnit!!, convUnit!!, value)
        }
    }
```

6. "strResult" method (function) will return a String value of a double value in parameter. There are 2 parameters:
    - double val, as an original value that will be converted to string.
    - Boolean rounded, as indicator. If true, the result must be rounded value with only 2 digits after decimal. If false, its mean no rounding process.

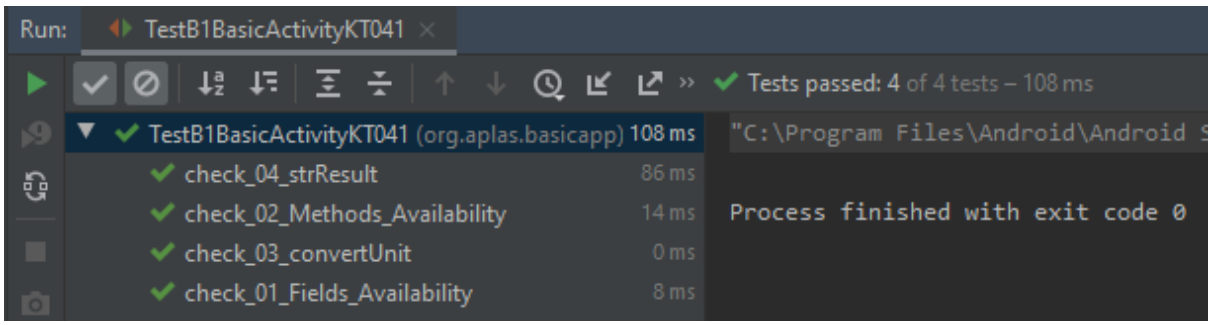
```
protected fun strResult(`val`: Double, rounded: Boolean):
String {
        return if (rounded) {
            val f = DecimalFormat("#.##")
            f.format(`val`)
        } else {
            val f = DecimalFormat("#.#####")
            f.format(`val`)
        }
    }
```

7. Copy “TestB1BasicActivityKT041.java” file to “org.aplas.basicapp (test)” folder. Replace if possible.



8. Right click on the “TestB1BasicActivityKT041.java” file then choose Run ‘TestB1BasicActivityKT041’ and click it. It may take long time to execute.

9. Get the result of your task. If passed you will get green check. If the test failed, you will get orange check get the messages and you must start your project again.



F. Testing.

You have to try until get all green checks and continue to the next task.