Mobile Dev "Mid"term

Name_____

1. 30 pts

Provide terms to match the following definitions. Spell correctly.

Type compatibility provided by the Kotlin "out" parameter on a type

Keyword used in Java equivalent declaration for "in"

Type of interface that is at the heart of how Kotlin does lambdas or function objects.

Kotlin keyword to declare a class similar to a C struct

Kotlin keyword to declare a singleton implementing some interface, like a Java anonymous inner class

Type rule by which a reference to Collection<X> can be assigned from a reference to Collection<Y> if Y is a base class of X.

Kotlin object that stands in for what would be static member data under other OO languages

Functional calling-pattern that replaces standard return values with lambdas that represent the entire remaining action of the caller after the call (full phrase, not acronym).

Min and max number of parameters a continuation may have

A UI class under JetPack that represents a small UI element that can be part of a larger Intent UI

Class that manages a number of coroutines, and provides common lifecycle control for them

Class that represents information in a model that tracks lifecycle-dependent listeners and updates them.

Kotlin sequence of information that is processed functionally and can be processed in parallel

Kotlin coroutine term for what would be a Promise in other languages.

2. 18 pts
Assume in the following code that fA, fB, etc. are all suspend functions.

```
launch(Dispatchers.Default) {
   println("Starting...")
   fA()
   fB()
   fC()
   println(fD())
}
```

a 6pts How many distinct continuation objects will result from the launch call by the time it has completed assuming standard CPS implementation? (Note, the answer is not 4.) Explain your answer.

b. 6pts How many distinct continuation objects will result from the way a Kotlin compiler probably implements this? (Again, not 4) Explain your answer.

c. 6pts Can fA perform operations on the UI, e.g. setting the text in a TextView? Why or why not?

3. 32 pts Revise this code to use CPS. Add no extra lines, and do not increase any line to > 80 chars. Build and compile your answer to demonstrate correctness.

```
var doPrint = {v: Int -> println("Value is $v")}
var max = {a: Int, b: Int -> if (a > b) a else b}

fun main() {
    val a = 42
    val  b = 43
    val c = 44
    doPrint(max(a, max(b, c)))
}
```

4. 28 pts The code below is a solution to the index-generation problem we did earlier in the term. Answer the following questions on it:

a. 8pts What type of collection, with what type of element, does the flatMapIndexed method return?

b. 8pts What is the type, including element type, of the lines parameter?

c. 4pts Why do we add 1 to it.second?

d. 8pts What does the filter call accomplish?

```
import kotlin.collections.*
import kotlin.text.Regex
import java.io.File

fun main(args: Array<String>) {
   File(args[0]).readLines()
    .flatMapIndexed({idx, line -> line.split(Regex("\\W+"))
    .filter({it.length > 0}).toSet().map({it to idx})})
    .groupBy({it.first}, {it.second + 1}).toSortedMap()
    .forEach({word, lines -> println("${word}: ${lines.joinToString(", ")}")})
}
```

5. 100 pts

Using Android Studio, make the following adjustments to the Quotes app from the end of our Layouts section. Demo for me as many of the following as you are able to complete.

a. 10pts Move the text to the upper left corner of the image

b. 30pts For each image, have two quotes, the second one placed at upper right. Adjust Quote to provide the second quote, and make all necessary changes to the code and layouts to support its display

c. 40pts Show the images in pairs, side-by-side, so that image 0 and 1 are the first pair, image 2 and 3 the second, etc. You may assume an even number of images. Make no changes to FixedSource. Change only QuoteAdapter and layout(s).

d. 20pts Make your work from part c able to accommodate an odd number of images by making the right side of the final pair non-displayed if the number of images is odd. (This may take a bit of research on visibility properties)

6. 120pts (extra credit -- only after completion of #5)

Adjust the code for Unscramble as indicated below. Demo for me as many of the following as you are able to complete.

a. 10pts Add a second "Hint" labelled TextView below that for the scrambled word.

b. 10pts Add a "Hint" button between Skip and Submit

c. 50pts Each time Hint is pressed, show one more letter of the answer in the Hint TextView, from left to right. Be sure you maintain the current number of hint letters (or equivalently the hint prefix) in the ViewModel, with appropriate ViewModel properties to support the hint display

d. 20pts When the final letter is revealed by a Hint buttonpress, behave exactly as if the Skip button has been pressed.

e. 30pts Accumulate score as a Double, displayed to two decimals, with each word earning a fractional score that is cut in half for each hint letter, e.g. .5 for one hint, .25 for a second, etc.