Written Exam

Programmering af mobile applikationer
Medialogy, 2nd semester

Friday 14 June 2024, 9.00 – 13.00

Navn: _____

Studienr: _____

Aalborg University | BSc in Medialogy | 2nd Semester

Programmering af mobile applikationer

Ordinary Examination

14 June 2024, 9.00 – 13.00

Instructions

- You have 4 hours to complete this examination.
- Neither electronic devices nor written material are allowed in the examination room.
- This examination consists of 8 questions. Each question is worth 10 marks. You must obtain at least 40 marks to pass.
- Do not write any answers on this question paper—answers written on the question paper will be ignored by the examiner. Write all your answers on the writing paper provided.
- Do not write your answers in pencil and do not use a pen with red or green ink. Use a pen with blue or black ink.
- Hand in no more than one answer to each question.
- Do not turn over until you are told to do so by the invigilator.

Question 1

a) Study the following code:

```
fun main() {
    val discountPercentage: Int = 0
    val offer: String = ""
    var item = "AAU t-shirt"
    discountPercentage = 20
    offer = "Sale! $discountPercentage% discount on $item!"

    println(offer)
}
```

This code contains two errors that prevent it from running.

i. Write down the line number(s) in which the errors occur. [2 marks]

ii. How would you correct these errors so that the program runs as intended? [2 marks]

iii. What would the corrected program print to the console? [1 mark]

b) Study the following code:

```
fun add(i: Int, j: Int) {
    println("$i + $j = ${i-j}")
}

fun main() {
    val i = 5
    val j = 2
    add(i,j)
}
```

What does this program print to the console window when it is run? [2 marks]

c) What does the following program print to the console window when it is run? [3 marks]

```
fun compare(i: Int, j: Int) {
    var d : Int = i - j
    var comparison: String = "equal to"
    if (d < 0)
        comparison = "less than"
    else if (d > 0)
        comparison = "greater than"
    println("$i is $comparison $j")
}

fun main() {
    compare(2,5)
    compare(3,-1)
    compare(0,0)
}
```

Question 2

a) Study the following code:

```
1  fun nums(i: Int) {
2      var s : String
3      when (i) {
4          in 1..5 -> s = "$i is between 1 and 5"
5          in 6..10 -> s = "$i is between 6 and 10"
6          1, 3, 5, 7, 9 -> s = "$i is an odd number between 1 and 10"
7          else -> s = "I don't know this number!"
8      }
9      println(s)
10 }
11
12 fun main() {
13     nums(4)
14     nums(7)
15     nums(11)
16 }
```

i. What does this program print to the console when it is run? [3 marks]

ii. If lines 5 and 6 were swapped, what would the program print to the console? [3 marks]

b) What does the following program print to the console window? [2 marks]

```
1  val krone: (Int) -> String = {"DKK$it"}
2
3  fun main() {
4      println("Parking costs ${krone(30)} per hour.")
5  }
```

c) What does the following program print to the console window? [2 marks]

```
1  fun main() {
2      var i: Int = 1
3      repeat(3){
4          repeat(2){
5              print(i)
6              i = i + 1
7          }
8      }
9  }
```

Question 3

a) What does the following program print to the console window? [4 marks]

```
class Point(val x: Int, val y: Int)

val points = listOf(
    Point(0,0), Point(0,1), Point(1,0), Point(1,1)
)

fun main() {
    points.forEach {
        println("Point(${it.x},${it.y})")
    }
}
```

b) What does the following program print to the console window? [4 marks]

```
class Point(val x: Double, val y: Double)

val points = listOf(
    Point(0.0,0.0), Point(0.0,1.0), Point(1.0,0.0), Point(1.0,2.0)
)

fun main() {
    println("(${points.fold(0.0) {sum, point -> sum + point.x}},"+
            "${points.fold(0.0) {sum, point -> sum + point.y}})")
}
```

c) Study the following program:

```
class A {
    object B {
        var x: Int = 1
        var y: Int = 2
    }
}

fun main() {
    println("${A.x},${A.y}")
}
```

This program will not run. However, by adding a single word at the beginning of line 2 of the program, it can be made to run and print out the following to the console:

1,2

What is the word that you need to add to the beginning of line 2 in order to make this program run? [2 marks]

Question 4

a) In the context of an Android app, what is a *configuration change*? [2 marks]
b) Give three examples of types of event that cause a configuration change. [3 marks]
c) In what sequence are the lifecycle callback methods executed in response to a configuration change? [3 marks]
d) What function do you need to use in order avoid losing data during a configuration change? [2]

Question 5

Study the following program and answer the questions that follow it.

```
1   import kotlinx.coroutines.*
2
3   fun main() = runBlocking {
4       launch {
5           delay(2000L)
6           println("Green")
7       }
8       launch {
9           delay(1000L)
10          println("Blue")
11      }
12      println("Red")
13  }
```

a. What does this program print to the console window when it is run? [3 marks]
b. Is each call to **launch** executed synchronously or asynchronously? [1 mark]
c. In which line or lines of the program is a *suspending function* called? [2 marks]
d. When a coroutine restarts after being suspended, will it necessarily be assigned to the same thread that it was running on before it was suspended? [1 mark]
e. What is the purpose of the code, "= runBlocking", in line 3? What would happen if it were removed? [3 marks]

Question 6

Suppose we have a database table called **email** that has the following columns in it: **id**, **subject**, **sender**, **folder**, **starred**, **read** and **received**. Write down SQL statements that do the following:

a) Select the **subject** and **sender** columns. [2 marks]
b) Count the total number of rows in the **email** table. [2 marks]
c) Get the maximum value in the **received** column. [2 marks]
d) Get a list of all the distinct values in the **sender** column. [2 marks]
e) Select all the rows in the **email** table where the value in the **folder** column is 'inbox'. [2 marks]

Question 7

a) WorkManager is an architecture component for background work that needs a combination of *opportunistic* and *guaranteed* execution. Explain what is meant by the terms "opportunistic execution" and "guaranteed execution" in this context. [2 marks]
b) Give an example of a task that would be a good use for WorkManager. [1 mark]
c) In the context of WorkManager, what is the difference between the Worker class and the CoroutineWorker class? [2 marks]
d) What is a WorkRequest object used for in the context of WorkManager? Name the two concrete implementations of this class that are provided by the WorkManager library. What is the difference between these two subclasses of WorkRequest? [5 marks]

Question 8

For each of the following statements, state whether it is true or false.

a) When building an app with Views, one constructs the UI inside a Layout.
b) Layouts are typically declared and defined using HTML.
c) A ConstraintLayout is a type of ViewGroup that lets you position and size Views using constraints.
d) A View is a special type of ViewGroup.
e) A ViewGroup can contain one or more Views, which may themselves be ViewGroups.
f) onCreateView is a lifecycle method of a Fragment that is called in order to have the Fragment instantiate its user interface View.
g) The onViewCreated() method is called after onCreateView() in the lifecycle of a Fragment.
h) onCreateView() is the recommended place to access and modify the Views within the layout.
i) You can access and set Views through a binding created by calling the bind() method of the FragmentEntryDialogBinding class.
j) When building an app with Views, a Fragment that hosts an XML layout replaces the concept of a Composable "screen".

END OF EXAMINATION