The ICT University

Spring 2024 Continuous Assessment №1

School of *ICT*

Course Code: SE 3242
Course Title: Android Application Development
Instructor: Engr. Daniel Moune

Instructions:

1. Your assignment should represent your own effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but **each student shall write down and submit his/her solutions separately**.

2. Please, include a cover page in your submission having your name, matricule, ICTU email and phone number.

3. Provide your answers using word-processing software then convert your document in **PDF** before uploading it to moodle.

4. Code listing should by typed and syntaxically coloured.

5. This assessment is made of **04** pages

March 26, 2024

Exercise 1: Kotlin Language Fundamentals

Provide an answer to the following questions. Questions requiring code should be provided with code listing having the appropriate color syntaxing features. Do not use screen capture from your editor to provide code for these questions

1. Dividing a program into functions
  a. is the key to object-oriented programming.
  b. makes the program easier to conceptualize.
  c. may reduce the size of the program.
  d. makes the program run faster.

2. A function name must be followed by _____.

3. A function body is delimited by _____.

4. Why is the main() function special?

5. A Kotlin instruction that tells the computer to do something is called a _____.

6. Write an example of an inline Kotlin comment and an example of a block comment.

7. An expression
  a. usually evaluates to a numerical value.
  b. indicates the emotional state of the program.
  c. always occurs outside a function.
  d. may be part of a statement.

8. Specify how many bytes are occupied by the following Kotlin data types in a 64-bits system:
  a. Type int
  b. Type long double
  c. Type float
  d. Type long

9. True or false: A variable of type char can hold the value 301.

10. What kind of program elements are the following?
  a. 12
  b. 'a'
  c. 4.28915
  d. JungleJim
  e. JungleJim()

11. Write Kotlin statements that display each of the following items along with their data types
  a. 'x'
  b. 'Jim'
  c. 2024

12. Write a Kotlin command that requests the name of the user as a string value, places it in a variable named 'name' then displays a greetings message with that name.

13. A relational operator
       a. assigns one operand to another.
       b. yields a Boolean result.
       c. compares two operands.
       d. logically combines two operands.

14. Name and describe the three flavors of for statement loops.

15. A variable defined within a block is visible
       a. from the point of definition onward in the program.
       b. from the point of definition onward in the function.
       c. from the point of definition onward in the block.
       d. throughout the function.

16. Write a *while loop* that displays the numbers from 100 to 110.

17. How many times is the loop body executed in a do loop?

18. Write a do loop that displays the first 100 *prime* numbers.

Exercise 2: Kotlin basic programming skills

Provide an answer to the following questions. For each question provide the code listing having the following color syntaxing theme (keywords in purple and literals in orange). Provide also a screen capture of the execution of your script in your favorite IDE. Use screen capture only to show the output of the execution of the code. Do not use screen capture to provide the code listing

1. Assuming there are 7.481 gallons in a cubic foot, write a Kotlin program that asks the user to enter a number of gallons, and then displays the equivalent in cubic feet.

2. Write a Kotlin program that generates the following table:(*use a single print statement to produce this output*)

2021	135
2022	7290
2023	11300
2024	16200

3. In the days of the old British empire, Great Britain used a monetary system based on pounds, shillings, and pence. There were 20 shillings to a pound, and 12 pence to a shilling. The notation for this old system used the pound sign, £, and two decimal points, so that, for example, £5.2.8 meant 5 pounds, 2 shillings, and 8 pence. (Pence is the plural of penny.) The new monetary system, introduced in the 1950s, consists of only pounds and pence, with 100 pence to a pound (like U.S. dollars and cents). We'll call this new system decimal pounds. Thus £5.2.8 in the old notation is £5.13 in decimal pounds (actually £5.1333333). Write a program to convert the old pounds-shillings-pence format to decimal pounds. An example of the user's interaction with the program would be

```
Enter pounds: 7
Enter shillings: 17
Enter pence: 9
Decimal pounds = £7.89
```

3. If you have two fractions, a/b and c/d, their sum can be obtained from the formula

```
  a       c        a*d + b*c
 --- +  ---   =    -----------
  b       d            b*d
```

For example, 1/4 plus 2/3 is

```
  1       2        1*3 + 4*2        3 + 8        11
 --- +  ---   =    -----------  =  -------   =  ----
  4       3            4*3            12          12
```

Write a Kotlin program named '*sum.kt*' that interacts with the user by asking to enter two fractions, and then displays their sum in fractional form. (You don't need to reduce it to lowest terms.) The interaction with the user might look like this:

```
Enter first fraction: 1/2
        Enter second fraction: 2/5
```

The output should be produced just like in the example provided above

4. Modify the Kotlin program of question 3 above. This time make sure the user can invoke the program from the command line with the fractions set as parameters. The program should not interact with the user but instead use the parameters provided in order to produce the output. The program invocation of the user might look like this:

```
> sum 1/4 2/3
```

The output should be produced just like in the example of question 3

```
    1       2        1*3 + 4*2        3 + 8          11
   --- +  ---   =   -----------   =   -------   =    ----
    4       3           4*3            12            12
```

5. Create a four-function calculator for fractions. For each of the each of the 04 arithmetic operations produce a Kotlin program based on the model of question 4. As a reminder here are the formulas for the four arithmetic operations applied to fractions

```
Addition:           a/b + c/d = (a*d + b*c) / (b*d)

Subtraction:        a/b - c/d = (a*d - b*c) / (b*d)

Multiplication:     a/b * c/d = (a*c) / (b*d)

Division:           a/b / c/d = (a*d) / (b*c)
```

The user should be able to invoke the program with the fractions as parameters and the output should produce the details just like in question 4

Amat Victoria Curam