

Partiel EFFICOM5 - Expertise développement mobile

1. Modalités de l'épreuve :

Uniquement ce quizz socrative sur votre ordinateur.

Ordinateur & IDE (IntelliJ, VSCode, Android Studio ...) :

☐ OUI

☒ NON

Accès Internet

☐ OUI

☒ NON

Supports de cours :

☐ OUI

☒ NON

Calculatrice :

☐ OUI

☒ NON

Dictionnaire :

☐ OUI

☒ NON

Conseils :

- Si pour la question de rédaction il vous faut plus de place, Ouvrez un notepad simple mais pas un IDE (par exemple word, notepad). Partagez le document en fin d'examen dans un zip a votre nom au surveillant de l'examen.

-Egalement afin d'éviter avec une déconnection toute perte de votre examen, merci d'avoir un bloc note en plus ou vous copiez vos réponses avant la soumission final de l'examen.

Bon courage.

Partie I - Questions de cours (30min - 41 Questions)

En Kotlin, quel est le mot clef pour déclarer une fonction ?

- ☒ (A) function
- ☐ (B) define
- ☐ (C) fun
- ☐ (D) declare

2. En Kotlin, la fin d'une instruction se termine par un point virgule (;) ?

- ☒ (T) True
- ☐ (F) False

3. En Kotlin, comment créer une variable de type numérique de valeur 5 ?

- ☐ (A) num = 5
- ☐ (B) val num = 5
- ☐ (C) int num = 5
- ☐ (D) num = 5 int
- ☐ (E) num : Int = 5

4. En Kotlin, val toto = "Hello" est modifiable ?

- ☐ (T) True
- ☐ (F) False

5. En Kotlin, var toto = "Hello" est modifiable ?

- ☐ (A) Vrai
- ☐ (B) Faux

6. En Kotlin, quel opérateur est utilisé pour fusionner 2 chaînes de caractères ?

- ☐ (A) le signe +
- ☐ (B) le signe &
- ☐ (C) le signe *
- ☐ (D) le mot clef "add"
- ☐ (E) la fonction ".plus()" de la class String

7. En Kotlin, comment peut-on créer une liste ?

- ☐ (A) myList:List {"value1","value2","value3"}
- ☐ (B) listOf() function
- ☐ (C) mutableListOf() function
- ☐ (D) myList["value1","value2","value3"]
- ☐ (E) none of those answers

8. En Kotlin, comment créer une plage de chiffres entre 5 et 15 ?

- ☐ (A) for (5.. 15)
- ☐ (B) for (5 to 15)
- ☐ (C) for (x in 5 to 15)
- ☐ (D) for (x in 5..15)

9. En Kotlin, est-il possible d'hériter des propriétés et fonctions d'une classe depuis une autre ?

- ☐ T True
- ☐ F False

10. En Kotlin, que signifie "fname" dans le prototype de fonction suivant : "fun myFunction(fname: String)"

- ☐ A Le parametre
- ☐ B La valeur de retour
- ☐ C Le type d'objet
- ☐ D aucune de ces propositions

11. En Kotlin, comment écrire correctement les paramètres d'entrée d'une fonction ?

- ☐ A fun sum(Int: ValueA, Int: valueB)
- ☐ B fun sum(Int ValueA, Int valueB)
- ☐ C fun sum(valueA: Int, valueB: Int)
- ☐ D fun sum(valueA Int, valueB Int)

12. En Kotlin, une fonction qui accepte en paramètre ou en retour une fonction est appelée ...

- ☐ A une extension
- ☐ B une lambda
- ☐ C Fonction d'ordre supérieur
- ☐ D Nullable receiver

13. En Kotlin, pour qu'une valeur ou une fonction d'une class puisse être hérité par une autre class il faut ajouter le mof clef ...

- ☐ A public
- ☐ B extends
- ☐ C open
- ☐ D final

14. En Kotlin, quelle est la syntaxe correcte pour convertir une chaine de caractères "42" en un Long

- ☐ A val l : Long = "42".toLong()
- ☐ B val l : Long = (Long)"42"
- ☐ C val l : Long = <Long>"42"
- ☐ D val l : Long = Long.parseLong("42")

15. En Kotlin, what is the difference between
var a: String? = "Kotlin"
var b: String = "Kotlin"

- ☐ (A) b est en final est ne peut pas être changé
- ☐ (B) a est volatile comme en java
- ☐ (C) a est en final est ne peut pas être changé
- ☐ (D) b ne peut jamais être null

16. En Kotlin, quelle(s) sont la/les déclarations de fonction valide(s) ?

- ☐ (A) function sum(a: Int, b: Int) : Int
- ☐ (B) fun sum(a: Int, b: Int): Int
- ☐ (C) int sum(a: Int, b: Int)
- ☐ (D) int sum(int a, int b)

17. En Kotlin, que fait l'opérateur "!!" ?

- ☐ (A) Il convertit n'importe quelle valeur en type non-null and renvoie une exception si la valeur est en fait "null"
- ☐ (B) Il compare si 2 valeurs sont identiques plutôt qu'égales
- ☐ (C) il renvoie l'opérande de gauche si elle n'est pas null sinon il renvoie l'opérande de droite.
- ☐ (D) C'est l'opérateur modulo en Kotlin, similaire à % en Java

18. En Kotlin, que ne propose pas la "data class" ?

- ☐ (A) Génération automatique des fonctions hashCode() et equals()
- ☐ (B) Génération automatique de la fonction toString()
- ☐ (C) Conversion automatique de/vers JSON
- ☐ (D) Génération automatique de la fonction copy(...), pour créer des copies d'instances

19. Quelle(s) mécanisme(s) permet aux applications Kotlin/Android de créer des tâches asynchrones (accès à une BDD, à une API Rest...) ?

- ☐ (A) une "Coroutine"
- ☐ (B) un Thread
- ☐ (C) un Fragment

20. Which components can be shared across platforms in Kotlin Multiplatform projects?

- ☐ (A) UI logic
- ☐ (B) Business logic
- ☐ (C) Database Datasource
- ☐ (D) Network Datasource

21. Name some popular libraries/frameworks compatible with Kotlin Multiplatform.

- ☐ (A) Ktor client
- ☐ (B) Kotlin serialization
- ☐ (C) SQLDelight
- ☐ (D) Kotlin coroutines
- ☐ (E) Kotlin Flows

22. How does Kotlin Multiplatform handle platform-specific implementations?

- ☐ (A) Through "expected" and actual declarations
- ☐ (B) "Expected" declarations define an interface or contract
- ☐ (C) "Implement" declarations provide platform-specific implementations
- ☐ (D) Through "expected" and "implement" declarations

23. What tools or IDE support is available for Kotlin Multiplatform development ?

- ☐ (A) IntelliJ IDEA with Kotlin plugin
- ☐ (B) Android Studio
- ☐ (C) Visual Studio Code with Kotlin extension
- ☐ (D) JetBrains Fleet

24. What tools or IDE support is available for Kotlin Multiplatform Compose development ?

- ☐ (A) IntelliJ IDEA with Kotlin plugin
- ☐ (B) Android Studio
- ☐ (C) Visual Studio Code with Kotlin extension
- ☐ (D) JetBrains Fleet

- 25.** What is the level of maturity or adoption of Kotlin Multiplatform in the development community?
- ☐ (A) Multiplatform is stable, ready for production
 - ☐ (B) Multiplatform is not stable, not for production
 - ☐ (C) Compose is not stable, not for production
 - ☐ (D) Compose is stable, ready for production
- 26.** Which of the following best describes the primary advantage of using Ktor's HttpClient?
- ☐ (A) Synchronous processing of HTTP requests
 - ☐ (B) Integration only with Android platforms
 - ☐ (C) Asynchronous and non-blocking by default
 - ☐ (D) Exclusive support for RESTful APIs
- 27.** How do you typically perform a GET request using Ktor's HttpClient?
- ☐ (A) `get()`
 - ☐ (B) `sendRequest(HttpMethod.Get)`
 - ☐ (C) `performGET()`
 - ☐ (D) `executeRequest(Method.GET)`
- 28.** What is the purpose of `MutableState` in Jetpack Compose?
- ☐ (A) To manage global application state
 - ☐ (B) To retain state within a composable and trigger recomposition
 - ☐ (C) To handle lifecycle events in composables
 - ☐ (D) To store read-only state
- 29.** Which function is used to create and manage mutable state in Jetpack Compose?
- ☐ (A) `stateOf()`
 - ☐ (B) `mutableStateOf()`
 - ☐ (C) `composeState()`
 - ☐ (D) `createState()`

30. What is Kotlin Flows

- ☐ (A) Flows is a multithreading framework library for kotlin
- ☐ (B) Flows are built on Kotlin coroutines, supporting sequential asynchronous processing
- ☐ (C) Flows are synchronous
- ☐ (D) Flows are limited to Android development only

31. Which operator is commonly used to transform data emitted by a Kotlin Flow?

- ☐ (A) map()
- ☐ (B) switchMap()
- ☐ (C) flatMap()
- ☐ (D) transform()

32. What role does `CoroutineScope` play in managing coroutines?

- ☐ (A) It determines the Thread used for asynchronous execution thanks to Dispatcher
- ☐ (B) It defines the lifecycle of the application
- ☐ (C) It specifies the UI thread for coroutine execution

33. Name some popular libraries/frameworks compatible with Kotlin Multiplatform.

- ☐ (A) Ktor
- ☐ (B) kotlinx.serialization
- ☐ (C) kotlinx.coroutines
- ☐ (D) KMock

34. How does Kotlin Multiplatform handle platform-specific implementations?

- ☐ (A) Through expected and actual declarations
- ☐ (B) Expected declarations define an interface or contract
- ☐ (C) Actual declarations provide platform-specific implementations
- ☐ (D) with suspend functions and coroutines

35. What tools or IDE support is available for KMP development?

- ☐ (A) IntelliJ IDEA
- ☐ (B) Android Studio
- ☐ (C) Visual Studio Code with Kotlin extension
- ☐ (D) JetBrains Fleet

- 36.** What is the level of maturity or adoption of Kotlin Multiplatform in the development community?
- ☐ (A) KMP is stable, ready for production
 - ☐ (B) KMP is not stable, not for production
 - ☐ (C) Compose Multiplatform is just few days ago stable, not ready for production
 - ☐ (D) Compose Multiplatform is stable, ready for production
- 37. Which of the following best describes the primary advantage of using Ktor's HttpClient?**
- ☐ (A) Synchronous processing of HTTP requests
 - ☐ (B) Integration only with Android platforms
 - ☐ (C) Asynchronous and non-blocking by default
 - ☐ (D) Exclusive support for RESTful APIs
- 38. How do you typically perform a GET request using Ktor's HttpClient?**
- ☐ (A) `get()`
 - ☐ (B) `sendRequest(HttpMethod.Get)`
 - ☐ (C) `performGET()`
 - ☐ (D) `executeRequest(Method.GET)`
- 39. What is the purpose of `MutableState` in Compose Multiplatform ?**
- ☐ (A) To manage global application state
 - ☐ (B) To retain state within a composable and trigger recomposition
 - ☐ (C) To handle lifecycle events in composables
 - ☐ (D) To store read-only state
- 40. Which function is used to create and manage mutable state in Compose?**
- ☐ (A) `stateOf()`
 - ☐ (B) `mutableStateOf()`
 - ☐ (C) `composeState()`
 - ☐ (D) `createState()`

41. What is Kotlin Flows ?

- (A) Flows is a multithreading framework library for kotlin
- (B) Flows are built on Kotlin coroutines, supporting sequential asynchronous processing of data on different architecture layers with emitters and collectors
- (C) Flows are synchronous
- (D) Flows are limited to Android development only

42. Partie II - Composables multiplateforme (60 min)

Description de la vue à créer :

La vue représente une application de gestion de tâches.
Elle se compose de trois sections principales :

En-tête de l'application :

- Affiche le titre de l'application.
- Un bouton pour ajouter une nouvelle tâche.

Liste de tâches :

- Affiche une liste de tâches avec leur titre et leur état (terminé ou non terminé).
- Chaque tâche est représentée sous forme de carte avec une case à cocher pour indiquer son état.

Section de détails de la tâche sélectionnée :

- Affiche les détails de la tâche sélectionnée (titre, description, date limite, etc.).

Coder les composables relatifs à cette vue et commentez votre organisation.

