1. **What is Inheritance? Try to explain in Mandarin.**

繼承在kotlin裡面指的是，讓現有的class去衍生出其他的class，而這些衍生的subclass可以使用其所繼承的(被衍生的)super class內的功能，即properties跟functions。另外，在sub class內除繼承的功能外也可以再覆寫或加入一些屬於自己的功能。

需要注意的是，在kotlin中所有的東西都被預設成了final，所以在被繼承的class和function前要加上"open"。

1. **How to split a string in Android? Explain your understanding, and even better to attach some sample code.**

在split function裡面有三種參數，分別是delimiter、ignorCase以及limit， delimiter用來作為string 進行split的標誌；ignoreCase用來忽略符合delimiter時的大小寫；limit用來預設可以return的substrings的最大數量。使用split會return List<String>。

以下以傳入兩個delimiters(“.”及”y”)、不忽略delimiters大小寫以及限制return數量為3作為演示：

1. fun main( ) {
2. var string = “str.str.strYstr.str”
3. println(string.split(".", "y", ignoreCase = false, limit = 3))
4. }
5. // outcome = [str, strYstr, str.str]
6. **What kind of key-value pair collection do we use in Android? Attach a sample code that uses for-loop to print each key-value pair.**

在kotlin裡面collection分為三種，分別是List、Set以及Map，它們各自又有mutable的類型。我們使用Map來做key-value的資料儲存。

以下使用for loop將Map裡面各個key及對應的value進行println：

1. fun main() {
2. val life = mapOf(
3. Pair("level.1", "Child"),
4. Pair("level.15", "Teenager"),
5. Pair("level.30", "Mage") )
6. val levelList = listOf<String>("level.1", "level.15", "level.30")
7. for (level in levelList) {
8. println(life.getValue(level))
9. }
10. }
11. // outcome = Child空行Teenager空行Mage
12. **What’s the difference between ArrayList and LinkedList? Try to explain in Mandarin.**

ArrayList是一種可改變長度的Array並 implement List；LinkedList是一種Doubly-linked list並implement List及Deque，其內部由一個個的資料項目組成，每個資料項目內部又由指標和資料本身構成。

以下列出三點區別：

1. LinkedList element在記憶體的位置是不連續的，而ArrayList element則是連續的。
2. 如同1)的區別所示，在操作上LinkedList比ArrayList快，因為當我們須新增或刪除資料時，LinkedList只需要改動這筆資料所連結的指標即可，然而ArrayList卻需要整個進行移動。
3. 在存取方面則是ArrayList較LinkedList快，因為array可以透過index取得資料，然而LinkedList必須從起點資料項目開始，跟著指標一路往後搜尋，即使搜尋的資料項目很接近起點依然如此。
4. **What are Value Type variables? List out the Value Type variables you know**

Value Type variable指的是variables中直接在記憶體儲存自己的資料，而非儲存引用的資料類型，與其相對的是Reference Type。

在網路上查找不太到Kotlin究竟是否擁有Value Type，相反的尋找到了Primitive Type的概念與其類似，其相反類型亦為Reference Type。推測Primitive Type為Value Type，而在Kotlin中最基本的Data Type屬於Primitive Type，有以下幾種：

1. Integer Data Type => Byte、Short、Int、Long
2. Floating-Point Data Type => Float、Double
3. Boolean Data Type => Boolean
4. **What’s the maximum value of Int? If you want to represent integers larger than the limit, what should you do instead of using int variables?**

Int的最大值是2,147,483,647，如果今天當值超過這個數字，則需要切換Int類型到Long，Long的最大值可以到達9,223,372,036,854,775,807。