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

多型是一個可以避免寫過多重複的程式碼的方式

用abstract抽象建立一個類別, 在用其他類別去extend這個父類別

可以更好的維護程式碼, 擴充性也更方便

最大的優點就是易於擴充及降低類別間的相依性

都是透過新增類別來擴充

2. Here are the 7 important lifecycle methods of an **Activity**:

Try to explain when they are called during the lifecycle of Activity.

i. onCreate()

程式經由onCreate()進入Created state, 在onCreate()應該要宣告並取得你會用到的UI元件, 以及UI元件的互動方法. ex. Click

ii. onStart()

接著, 程式會經由onStart()進入 Started state, 在onStart()應該要取得使用者的個人使用資料, 並且取得部分需要的資源, ex. GPS.

iii. onResume()

等程式進入onResumed state, 就是進入了螢幕的顯示狀態, 所以在onResume()的方法中必須取得必要的顯示資源, ex. Camera

iv. onPause()

如果程式跳離了畫面但上一個activity還看得到(也就是部分看得到), 那activity就會進入Paused state

v. onStop()

如果整個跳離了(完全看不到), 則activity會進入 Stopped state.

vi. onDestroy()

最後完全結束Activity之前，呼叫onDestroy方法

3. What is the **Android Jetpack**?

Android Jetpack 是一組組件、工具和架構指南，可幫助快速輕鬆地構建出色的 Android 應用程序。組件可以單獨採用，但可以協同工作，同時利用 Kotlin 語言功能提高您的工作效率。Jetpack 管理繁瑣的活動，如後台任務、導航和生命週期管理，因此您可以消除樣板代碼並專注於使您的應用程序出色的因素。Jetpack 組件圍繞現代設計實踐構建，具有向後兼容性，可減少崩潰和內存洩漏。

4. What is Coroutines? Why do we use it? Try to explain in Mandarin.

Coroutines(協程), 計算機程序組件, 通過允許任務掛起和恢復執行, 來支持非搶佔式的多任務

我的想法是, 他是一個可以讓很多程式碼同時進行的資料庫, 可以透過

suspend function的delay 來延遲執行程式碼,

launch返回Job, 代表一個協程, 我們可以用Job的join()方法來顯式地等待這個協程結束

Job還有一個重要的用途是cancel(), 用於取消不再需要的協程任務. 等還有很多方法可以呼叫