**Batch #13 / Android Class**

**Remote Learning Assignment - Week 4**

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

多型 (Polymorphism)的設計特性能夠簡化程式碼結構，能在一組類別裡複用個性化函數。多型使用同一個操作介面，以操作不同的物件，能讓程式對個別物件的依賴降低，更容易修改、維護、增加調整彈性度。

1. **Here are the 7 important lifecycle methods of an Activity.Try to explain when are they called during the lifecycle of Activity.**
2. onCreate()

在系統首次創建Activity時會callback,並進入onCreate狀態

1. onStart()

在系統要進入onStart時會callback onStart()，使Activity進入可見的狀態

1. onResume()

在系統要進onResume時或從onStop返回onResume狀態時會callback，使應用能與用戶進行互動，維持focus的狀態

1. onPause()

當發生中斷事件，Activity要進入onStop狀態時，系統會callback，表示Activity不在前台，但用戶處於多窗口模式時Activity仍然可見

1. onStop()
2. 當Activity不再可見，表示系統已進入onStop狀態，此時會callback onStop()
3. onDestroy()

銷毀Ativity之前，系統會callback onDestroy()，因為Activity即將結束；或因為配置變更（例如螢幕旋轉或多窗口模式），系統暫時銷毀Activity。

1. **What is the Android Jetpack?**
2. Jetpack 是一個由多種library組成的套件，協助開發者遵循最佳做法、減少樣板程式碼及編寫可運行於Android 各版本及設備的程式碼，讓開發者能夠更專注於編寫程式的工具。
3. Jetpack 優點

* 遵循最佳做法：Android Jetpack 組件採用最新的設計方法構建，具有向後兼容性，可以減少崩潰和內存洩露。
* 消除樣板代碼：Android Jetpack 可以管理各種繁瑣的Activity（如後台任務、導航和生命週期管理），以便專注於打造出色的應用。
* 減少不一致：這些library可在各種Android 版本和設備中以一致的方式運作，降低複雜性。

1. Android Jetpack 組件覆蓋以下 4 個方面：基礎（Foundation）、架構（Architecture）、行為（Behavior） 、界面（UI）。
2. **What is Coroutines? Why we use it? Try to explain in Mandarin.**
3. Coroutines表示協程，是一種並行的設計模式，可以用來簡化異步執行的代碼，有助於管理長時間運行作業，避免阻塞主線程及導致APP無回應。
4. Coroutines優點

* 輕量：使用suspend，用於暫停執行當前協程，避免阻塞主線程
* 較少內存洩漏：scope，設定協程作用的範圍，可以是Main thread 或是 IO thread
* 內建cancel()取消功能
* Jetpack集成