***Q1***

**Polymorphism-多型**

指子類別繼承父類別時，同時改寫父類別的函式或變數，而我們可以宣告一個屬於父類別的變數，去存取子類別中改寫自父類別的函式或變數。若有一個父類別是動物，那麼有一個子類別鳥繼承自動物這個類別，當我們看到鳥的時候，可以說牠是一隻鳥，也可以說牠是一隻動物。同樣的東西，卻有不同的表示型態。

***Q2***

**onCreate()**

只會在 Activity 被產生時呼叫一次，通常用來做程式的初使化動作，一次性的功能通常會放在onCreate()

**onStart()**

當Activity變得可見時呼叫

**onResume()**

每次進到 Activity 都會被呼叫，中間因為一些操作過程離開目前的 Activity，之後又回到這個 Activity 的話，onResume() 也會再次被呼叫

**onPause()**

當 Activity 要進入背景時，在 Android 中通常只會有一個 Activity 在最上方，其餘不在最上方的Activity 都屬於存在背景中，當目前的 Activity 被另外一個 Activity 取代時，目前的 Activity 就會進入背景並且呼叫 onPause()

**onStop()**

當Activity變得不可見時呼叫

**onDestroy()**

當 Activity 要整個被完全結束掉時被呼叫

**onRestart()**

當一個Activity再次啟動之前會被呼叫

***Q3***

2018年Google I/O 發佈的一系列輔助android開發者的實用工具，合稱 **Android Jetpack**。

**Android Jetpack** 包含4個部分的組件: Architecture / Foundation / Behavior / UI 。

裡面是大量專為 Android 開發而設的資料庫 (Library) ，它們可以全面的幫助開發者掌掃 Background task 、導向(Navigation) 以及 Lifecycle management ，這些都是開發者需要處理的底層問題。

***Q4***

**Activity**

[Activity](https://developer.android.com/reference/android/app/Activity.html?hl=zh-TW) 是提供畫面的應用程式元件，使用者可以與此畫面互動以執行動作，例如撥號、拍照、傳送電子郵件或檢視地圖。 每個 Activity 都會有專屬視窗，用於繪製其使用者介面。視窗一般會占滿螢幕，但也可能小於螢幕或在其他視窗上方浮動。

**Service**

[Service](https://developer.android.com/reference/android/app/Service.html) 是可以在背景中長時間執行操作的應用程式元件，且不提供使用者介面。 也是另一個可以啟動服務的應用程式元件，就算使用者切換至其他應用程式，也會繼續在背景中執行。

**Content Provider**

內容供應程式可管理中央資料存放庫的存取權。供應程式是 Android 應用程式的一部分，通常可提供本身的 UI 方便使用者處理資料。 不過，內容供應程式主要是供其他應用程式使用 (透過供應程式用戶端物件進行存取)。 供應程式與供應程式用戶端可提供一致的標準介面，除了可用於存取資料，還能用來處理程序間通訊以及保護資料存取的安全。

**Broadcast Receiver**

是一種系統級的全範圍通知，而不是針對單一位置，這樣的好處是，我們不必明確指出需要回應這個廣播的位置，就可以驅動需要回應此廣播的事件，在 android 中，我們時常需要利用此一機制，讓程式能在任何時候，對於不同的廣播訊息做出正確的反饋。

**Notification**

通知泛指 Android 在您應用程式 UI 以外的位置所顯示的訊息，可為使用者提供提醒內容、其他使用者傳來的通訊內容，或是來自您應用程式的即時資訊。使用者只要輕觸通知即可開啟應用程式，也可以直接透過通知執行相關動作。

Runtime Permission

**ANR - Application Not Responding**

在Android上，如果應用程式有一段時間響應不夠靈敏，系統會向使用者顯示一個對話方塊，這個對話方塊稱作應用程式無響應（ANR：Application Not Responding）對話方塊。使用者可以選擇「等待”」而讓程式繼續執行，也可以選擇「強制關閉」。所以一個流暢的合理的應用程式中不能出現ANR，而讓使用者每次都要處理這個對話方塊。因此，在程式裡對響應效能的設計很重要，這樣系統不會顯示ANR給使用者。

**OOM -** [**Out Of Memory**](http://stephenandroiddv.blogspot.com/2017/10/androidout-of-memoryoom.html)

在Android中，一個Process 只能使用16M記憶體，要是超過了這個限定就會跳出這個異常。

**Exception**

java將所有的錯誤封裝為一個對象，其根本父類為Throwable, Throwable有兩個子類：Error和Exception。

Exception是Throwable的一個主要子類。 Exception下面還有子類，其中一部分子類分別對應於Java程序運行時常常遇到的各種異常的處理，Exception子類下面的另一部分子類對應於Java程序中的非運行時異常的處理，這些異常也稱為顯式異常非RumtimeException。它們都是在程序中用語句拋出、並且也是用語句進行捕獲的，比如，文件沒找到引起的異常、類沒找到引起的異常等。

**Material Design**

是 Google 公佈的一套同時適用於Android、iOS、Web 等各種平台，同時能適用於手機、平板、電視、電腦螢幕等不同裝置的跨平台/裝置的設計規則（仍需為各裝置設計，但鼓勵共用更多相同元素）。

***Q5***

**Coroutine - 協作程序**

將 function 、method 協同其他更多的 function、method 共同作業，這件事情稱為 Coroutine。允許 method 被暫停( suspended)執行之後再回復(resumed)執行，而暫停執行的 method 狀態允許被保留，復原後再以暫停時的狀態繼續執行。需要長時間多線程(Multithreading)運行時使用。