1. What is ​Encapsulation​? Try to list out the benefits of Encapsulation.
2. 建立類別需要決定類別函數及類別屬性是否可以被外界取用，若不需要被外界所取用的項目，即可加入可見性修飾符號將其隱藏，是為封裝 (Encapsulation)，例如：private、protected、internal等
3. Encapsulation優點

* 隱藏訊息,可以接觸到該物件的程式碼是有限的,即可減少錯誤的發生
* 禁止外界接觸,限制使用方式,讓程式開發者更容易使用、測試及debug

1. Try to explain what is ​enum​ and why we use them.

Enum(列舉)​ ：只有固定物件的類別，可以採用列舉的概念來設計，先定義好各個物件，即可直接利用。

1. How to use ​enum​ in Kotlin? Attach a sample code that use enum in Kotlin.

列舉常數以逗號分隔，每個列舉常數都是一個物件；另外，列舉類別提供內建的屬性與方法，能取得列舉物件的名稱，或藉由名稱字串來取得對應的物件

* 「name」屬性：代表列舉物件的名稱。使用「valueOf」方法，可傳入字串，並取得名稱相符的物件（大小寫需相符）。若物件不存在，會拋出例外。另外，「toString」方法預設會回傳name屬性，因此也能自行使用或覆寫。
* 「ordinal」屬性：代表該物件在類別中的順序，就像陣列與List的索引。

enum class Fruit(val chineseName: String, val Price:Int) {

APPLE("蘋果", 10),

BANANA("香蕉", 20),

GUAVA("芭樂", 40)

}

fun main(args: Array<String>) {

println(Fruit.APPLE.toString())

}

1. What are the ​differences​ between ​LinearLayout and​RelativeLayout​? Try to explain in details.

* LinearLayout中的所有 View 都是線性排列的，排列上的限制性較高，參數設定較少且單純，可分為兩種方向依序排列:垂直方向(Vertical)及水平方向(Horizontal)
* RelativeLayout中的所有 View 都是有相對關係的，如 View 與 View 之間或 View 與 Layout 之間有相對關係，較處理各種解析度的狀況，佈局上相當靈活，然而參數設定也較為複雜。

1. Try to explain the benefits of ​ConstraintLayout​. Why you should use it?

ConstraintLayout 是透過物件本身與其它物件之間的約束來決定它的位置，物件上下左右的鍊子就是它的約束，處是在做一些較複雜的界面時，可以用各種約束來定位自已的位置，不用使用多層 layout 嵌套來處理，ConstraintLayout 取代傳統 Layout 的複雜組合，把 xml 檔變得更加扁平，效率很容易就能提升利用，新增功能如(1)利用 Guideline 來定位(2)透過 bias 對 view 的位置做一些偏移(3)用 ratio 的方式幫 view 調整大小(4)引入 chain 的概念，幫一組 views 做 Layout