

基礎行動裝置應用程式設計

Android 6 範例試卷

【 認證說明與注意事項 】

- 一、本項考試為操作題，所需總時間為 60 分鐘，時間結束前需完成所有考試動作。成績計算滿分為 100 分，合格分數為 70 分。
- 二、操作題為兩大題，每題 50 分，共計 100 分。
- 三、操作題所需的檔案皆於 C:\ANS.CSF 資料夾內讀取。題目存檔方式，請依題目指示儲存於 C:\ANS.CSF 資料夾，測驗結束前必須自行存檔，並關閉 Android Studio，檔案名稱錯誤或未自行存檔者，均不予計分。
- 四、操作題每大題之各評分點彼此均有相互關聯，作答不完整，將影響各評分點之得分，請特別注意。題意內未要求修改之設定值，以原始設定為準，不需另設。
- 五、試卷內 0 為阿拉伯數字，O 為英文字母，作答時請先確認。所有滑鼠左右鍵位之訂定，以右手操作方式為準，操作者請自行對應鍵位。
- 六、有問題請舉手發問，切勿私下交談。

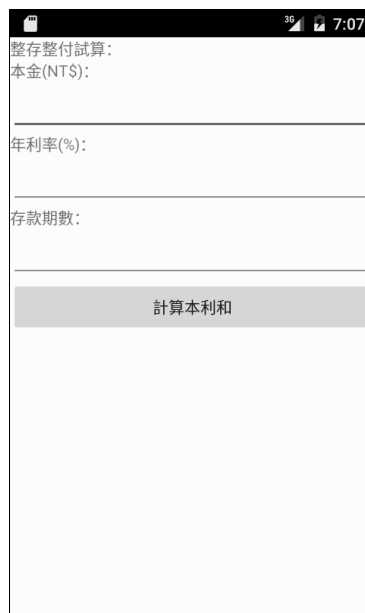
壹、操作題 100% (第一題至第二題每題 50 分)

請依照試卷指示作答並存檔，時間結束前必須完全跳離 Android Studio。

一、整存整付計算機

1. 題目說明：

請開啓 **COM.TQC.GDD01** 專案，設計「整存整付計算機」應用程式。在輸入本金、年利率與存款期數後，按下「計算本利和」按鈕，即可在下方顯示本利和，請依下列題意進行作答，作答完成請儲存所有檔案。



The image shows a mobile application interface for calculating the total amount of a fixed deposit. It features a title bar with the text '整存整付試算:'. Below the title bar are three input fields: '本金(NT\$):', '年利率(%):', and '存款期數:'. A button labeled '計算本利和' is positioned below the input fields. The interface is displayed on a mobile device screen with a status bar at the top showing the time as 7:07.

<參考圖>

2. 設計說明：

- (1) 請開啓 `main.xml` 檔案，依參考圖利用 `LinearLayout` 製作出整存整付計算機的主要畫面，並開啓 `strings.xml` 檔案，將設計說明(2)所需的字串常數先定義於此。
- (2) 請依參考圖設計四個 `TextView`，修改設定以設計說明(1)所定義的字串常數，使其分別顯示以下文字：【整存整付試算：】、【本金(NT\$):】、【年利率(%):】、【存款期數：】。
- (3) 請依參考圖設計三個 `EditText`，分別置於【本金(NT\$):】、【年利率(%):】、【存款期數：】 `TextView` 下方，提供輸入相對應的資料。
- (4) 三個 `EditText` 皆限制僅能輸入數值（整數）。

- (5) 請設計一個 Button，string 值設定為【計算本利和】。
- (6) 在 Button 下方，需再加入一個顯示本利和計算結果的 TextView，text 值預設為空，程式起始時不顯示。
- (7) 點選「計算本利和」按鈕，進行本利和總數計算，將總額顯示於下方 TextView 元件中，顯示【本利和為：xx】，將計算出的金額代入 xx 內，金額需顯示為整數。
 - ❖ 提示：整存整付公式：本利和=本金*(月利率 + 1)^存款期數。

3. 執行結果參考畫面：



4. 自行測試程式是否達成下列結果：

- (1) 如參考圖所示，畫面元件依 res/layout/main.xml 的定義正確顯示。
- (2) 開啓 strings.xml 檔案，將字串常數值設定於此。
- (3) 三個 EditText 元件需限制僅能輸入數值，且數值為整數。
- (4) 顯示本利和計算結果的 TextView，text 值預設為空，程式起始時不顯示。
- (5) 點選「計算本利和」的按鈕，依公式計算出本利和總數。

5. 評分項目：

項	目	配 分	得 分
(1)	畫面元件依 res/layout/main.xml 的定義正確顯示，各個物件及其屬性設定與參考圖相同	20	
(2)	字串常數設定於 strings.xml 檔案內	7	
(3)	三個 EditText 元件需限制僅能輸入數值，且數值為整數	8	
(4)	顯示本利和計算結果的 TextView，text 值預設為空，程式起始時不顯示	3	
(5)	點選「計算本利和」的按鈕，依公式正確計算出本利和總數	12	
	總 分	50	

二、點餐系統

1. 題目說明：

請開啓 **COM.TQC.GDD01** 專案，設計「點餐系統」應用程式。按下「進入點餐系統」的 Button 時，跳出 AlertDialog 呈現菜單列表，每樣餐點旁並有 CheckBox 提供勾選並可再次修改，請依下列題意進行作答，作答完成請儲存所有檔案。



<參考圖>

2. 設計說明：

- (1) main.xml 為 Activity 所使用的 Layout，已設計好程式內所需使用的元件。
- (2) strings.xml 中已定義好程式內所需使用的文字。
- (3) GDD01.java 內已宣告一字串陣列 s1，陣列包含所有餐點的名稱共 5 筆。
- (4) 開啓 GDD01.java，實作 Button 的 OnClickListener，在按下「進入點餐系統」Button 時，跳出 AlertDialog 呈現菜單列表，設定以下項目，顯示如第 3 點執行結果參考畫面項目中的第(1)小項所示。

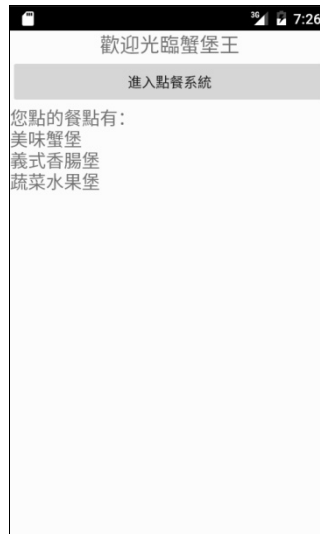
- a. 設定 AlertDialog 的 Title 為已設定在 strings.xml 中的字串 str2，顯示【您點的餐點有：】。
 - b. 將字串陣列 s1 內的餐點以列表方式呈現於 AlertDialog 中，且每樣餐點旁都有 CheckBox 可以勾選。
 - c. 為 AlertDialog 加入「確認」與「離開」兩個 Button 控制項。
- (5) 選擇完畢後按下「確認」按鈕，會將點餐的選取結果顯示於畫面下方的 TextView 裡。
- (6) 點選「離開」按鈕，將 AlertDialog 關閉。
- (7) 再次按下「進入點餐系統」的 Button 時，跳出 AlertDialog，並將原先已點的餐點帶入 AlertDialog 中，已點餐點旁的 CheckBox 會顯示已勾選的狀態並可再次修改，顯示修改後的餐點於畫面下方的 TextView 裡。

3. 執行結果參考畫面：

- (1) AlertDialog 呈現菜單列表，每樣餐點旁都有 CheckBox 可以勾選，並於 AlertDialog 加入「確認」與「離開」兩個 Button 控制項。



(2) 選擇完畢後按下確認按鈕，會將點餐的選取結果顯示於畫面下方的 TextView 裡。



(3) 再次按下「進入點餐系統」的 Button 控制項時，會再次跳出 AlertDialog，並將原先已點的餐點帶入 AlertDialog 中，已點餐點旁的 CheckBox 會顯示已勾選的狀態並可再次修改。



(4) 進行再次修改，顯示修改後的餐點於畫面下方的 TextView 裡。



4. 自行測試程式是否達成下列結果：

- (1) 按下「進入點餐系統」的 Button 時，跳出 AlertDialog。
- (2) 設定 AlertDialog 的 Title 為已設定在 strings.xml 中的字串 str2，顯示【您點的餐點有：】。
- (3) AlertDialog 中顯示餐點列表，且餐點旁有 CheckBox 可以勾選。
- (4) AlertDialog 有「確認」與「離開」兩個 Button 控制項。
- (5) 按下「確認」按鈕時，將選擇的餐點顯示在畫面中的 TextView 裡。
- (6) 按下「離開」按鈕時，直接將 AlertDialog 關閉。
- (7) 再次按下「進入點餐系統」Button 控制項時，會將已點的餐點資料帶入，並將該餐點的 CheckBox 設定為已勾選，且可再次進行修改，並重新顯示所選擇之餐點。

5. 評分項目：

項 目	配 分	得 分
(1) 按下「進入點餐系統」的 Button 時，跳出 AlertDialog	5	
(2) 設定 AlertDialog 的 Title 為已設定在 strings.xml 中的字串 str2，顯示【您點的餐點有：】	3	
(3) AlertDialog 中顯示餐點列表，且餐點旁有 CheckBox 可以勾選	10	
(4) AlertDialog 有「確認」與「離開」兩個 Button 控制項	5	
(5) 按下「確認」按鈕時，將選擇的餐點顯示在畫面中的 TextView 裡	12	
(6) 按下「離開」按鈕時，直接將 AlertDialog 關閉	3	
(7) 再次按下「進入點餐系統」Button 控制項時，會將已點的餐點資料帶入，並將該餐點的 CheckBox 設定為已勾選，且可再次進行修改，並重新顯示所選擇之餐點	12	
總 分	50	