

學號：1234567
姓名：王小明
作業撰寫時間：180 (mins, 包含程式撰寫時間)
最後撰寫文件日期：2023/09/22

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

```

sequenceDiagram
    participant A as :InstanceA
    participant B as :InstanceB
    participant C as :InstanceC

    A->>A: 
    activate A
    A->>B: 
    activate B
    B->>C: 
    activate C
    C-->>B: 
    deactivate C
    B-->>A: 
    deactivate B
    A-->>A: 
    deactivate A
  
```

UML sequence diagram illustrating object creation and message passing between three instances: `:InstanceA`, `:InstanceB`, and `:InstanceC`.

The diagram shows the following sequence of events:

- `:InstanceA` receives an external message (solid arrow).
- `:InstanceA` sends a message to `:InstanceB` (solid arrow).
- `:InstanceB` sends a message to `:InstanceC` (solid arrow).
- `:InstanceC` returns a message to `:InstanceB` (dashed arrow).
- `:InstanceB` returns a message to `:InstanceA` (dashed arrow).
- `:InstanceA` returns a message to the external caller (dashed arrow).

Callouts provide additional context:

- 回傳物件 訊息生命線** (Return object message lifeline): Points to the dashed return arrows.
- 程式中被建立的物件 執行規格長條格 訊息呼叫(操作)** (Object created in program, execution specification bar, message call/operation): Points to the solid activation bars and message arrows.

在畫循序圖時，需要考慮系統的整體流程，確保循序圖能夠清晰地展示系統中的各個部分之間的交互作用。長方形代表生命線方格，下邊虛線稱為生命線，代表物件存在的時間長度，以類似柵欄的方式描述物件間的互動。每一個新的物件就加入到圖的右邊，表示出訊息呼叫的次序。