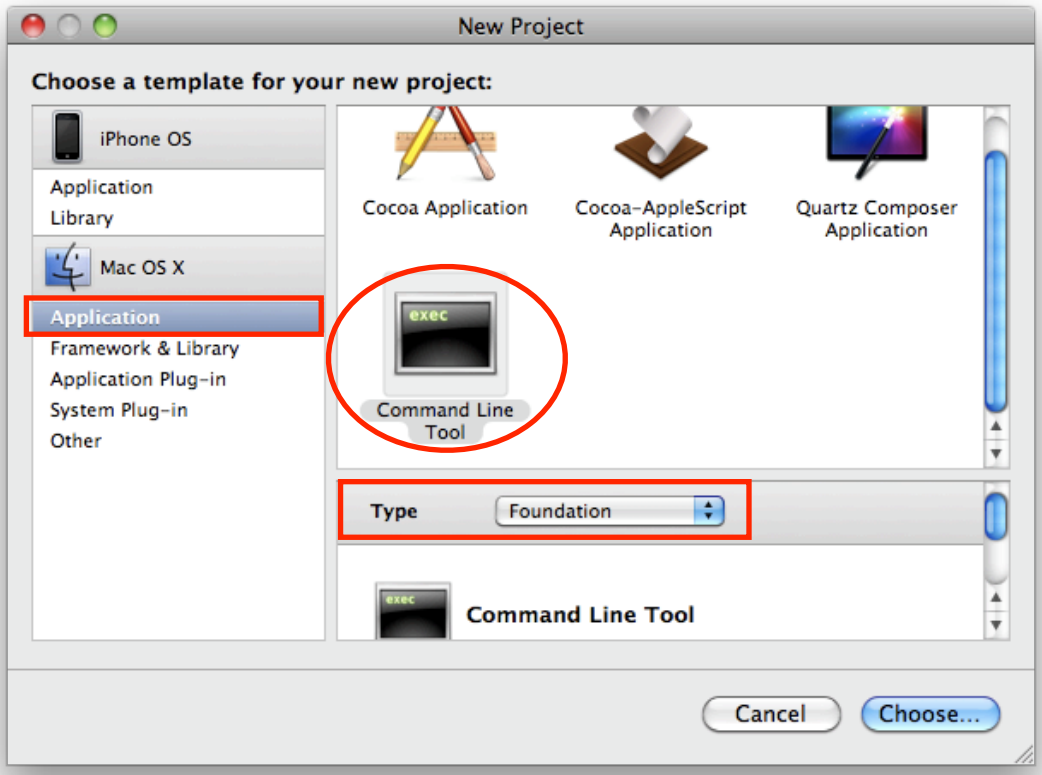
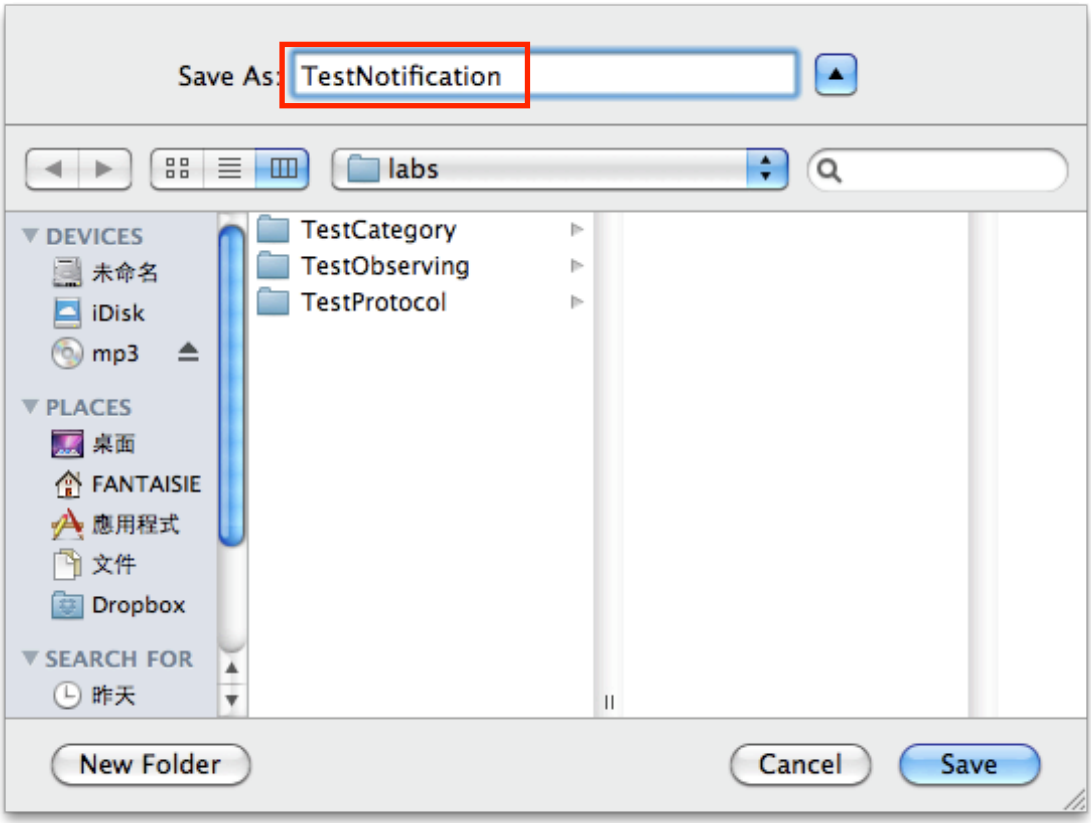
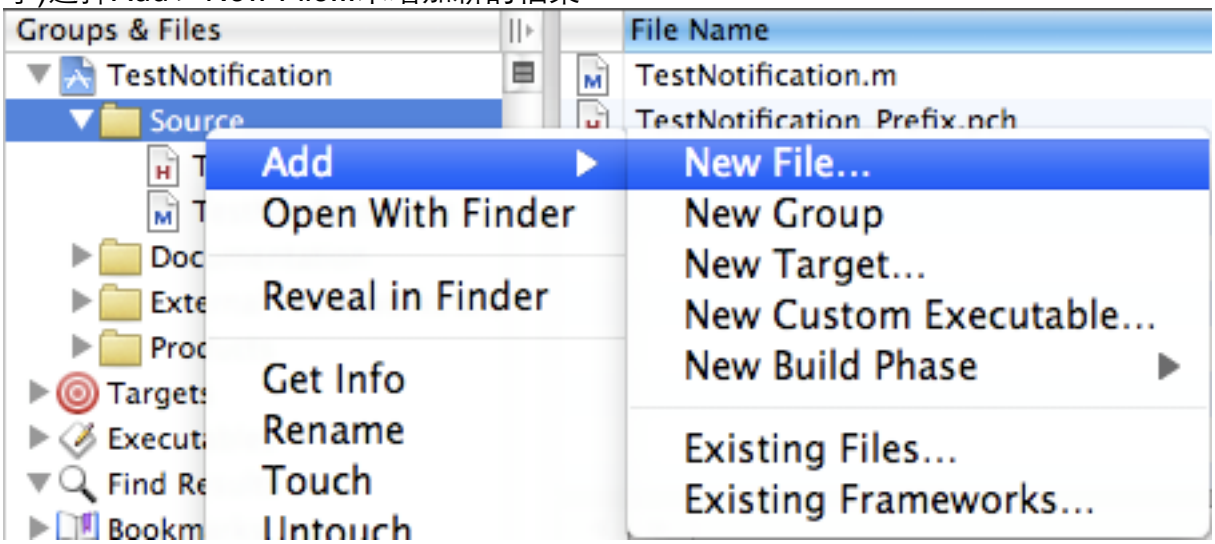
Lab TestNotification

Step1. 在File開啓一個新的project, 選擇 MAC OS X的Command line Tool, Type選擇 Foundation, 將project命名為 TestNotification

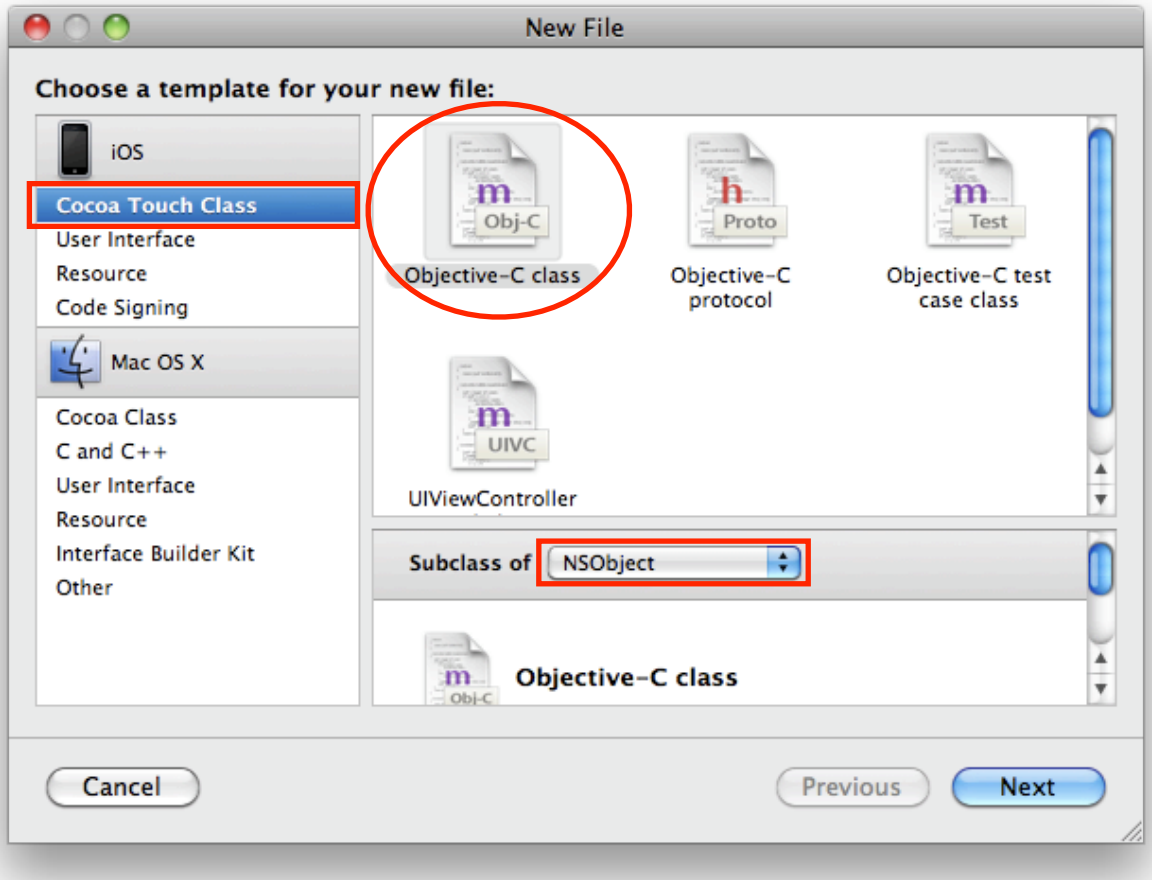




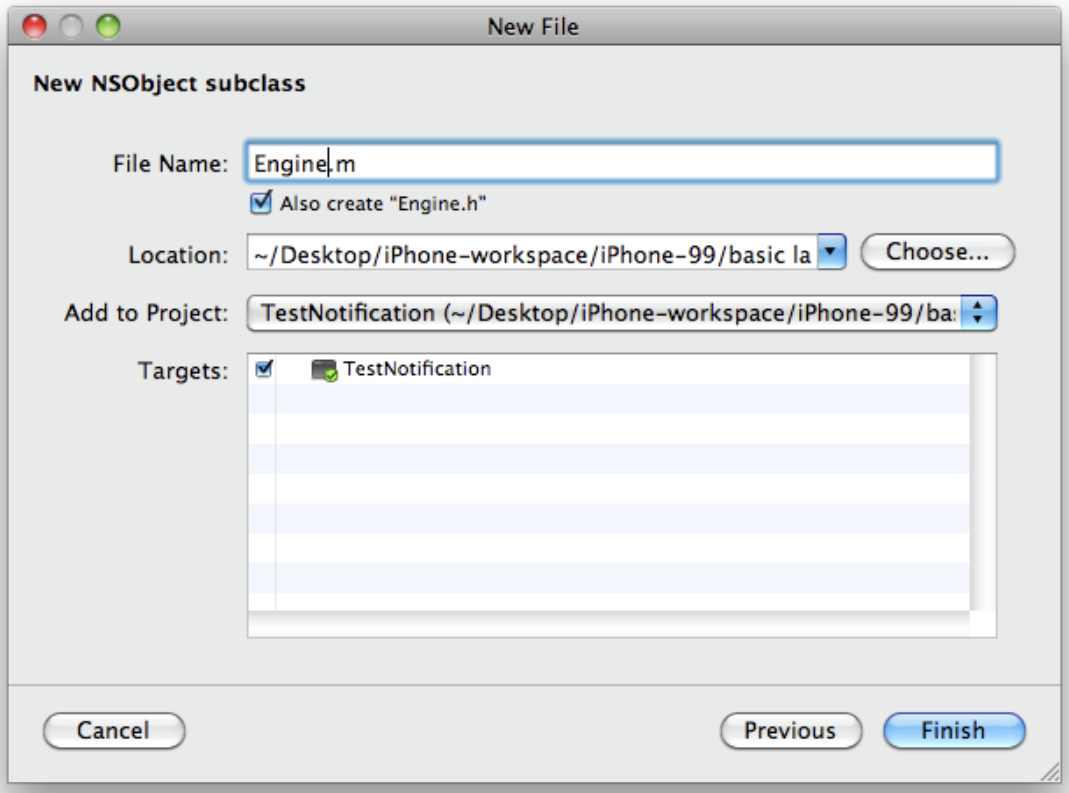
Step2. 在Xcode左邊Groups & Files 視窗中, 在Source這個路徑下點右鍵(若無滑鼠ctrl+點擊)選擇Add > New File...來增加新的檔案



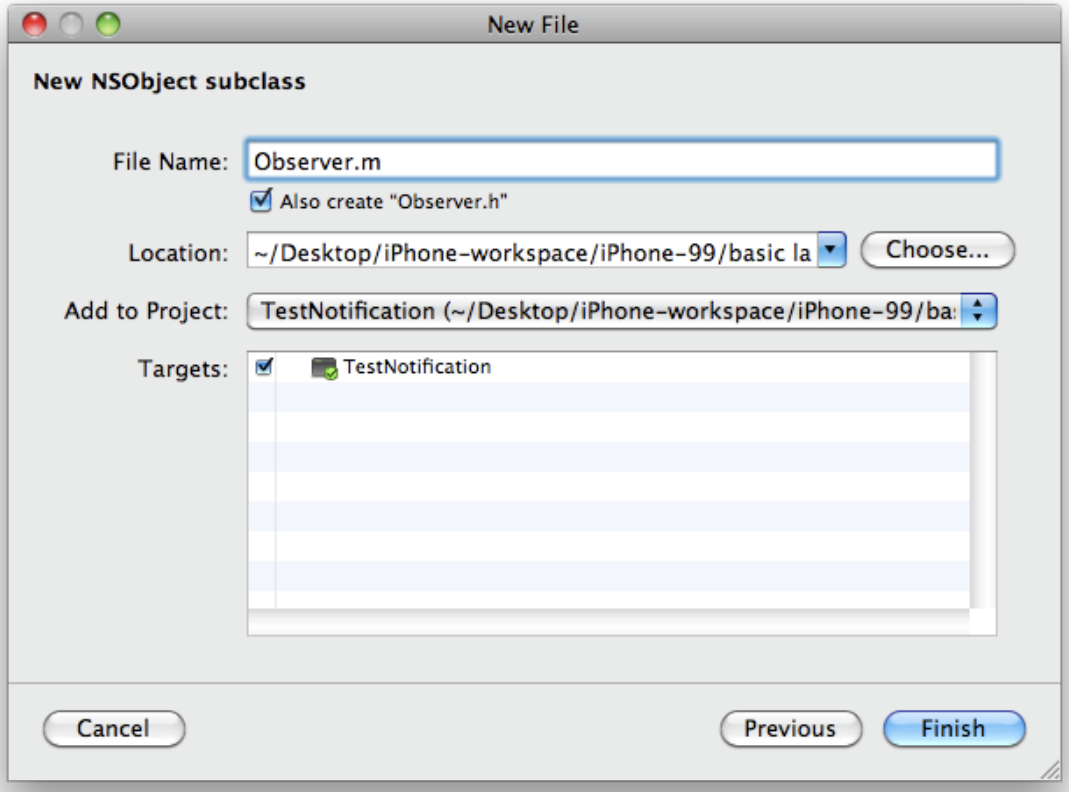
Step3. 選擇iOS裡的Coca Touch Class裡的, 並選擇Subclass of **NSObject**下方有敘述這個Objective-C有includes <Foundation/Foundation.h> 這個標頭檔



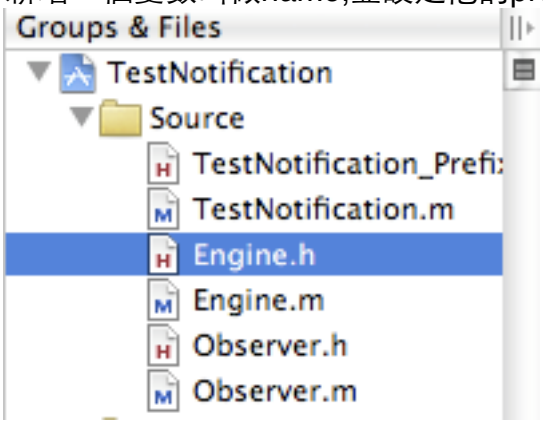
Step4. 將新增的File命名為Engine.m, 記得勾選Also create “Engine.h”



Step5. 依照上面的方式,再新增一個class叫做Obsever



Step6. 在Xcode左邊Groups & Files 視窗中,開啓Source > Engine.h
新增一個變數叫做name,並設定他的property, 並宣告一個method叫做setName:




```
#import <Foundation/Foundation.h>


@interface Engine : NSObject {
    NSString *name;
}

@property (retain) NSString * name;
-(void) setName:(NSString *) n;

@end
```

Step7. 開啓Engine.m, 對name的property做相對應的synthesize, 並實作setName:這個method, 除了設定name以外,還會發出一個Notification叫做”NameUpdated”

```
#import "Engine.h"


@implementation Engine

@synthesize name;

-(void) setName:(NSString *) n{
    name = n;
    [[NSNotificationCenter defaultCenter]
postNotificationName:@"NameUpdated" object:self];
}

@end
```

Step8. 開啓Observer.h,宣告三個method, 一個是增加observing,另一個是移除observing, 還有一個是當觀察判斷到時為執行的method

```
#import <Foundation/Foundation.h>


@interface Observer : NSObject {

}
- (void) addObserving:(NSObject *) observedObject;
-(void) removeObserving;
-(void) gotNote:(NSNotification *)note;
@end
```

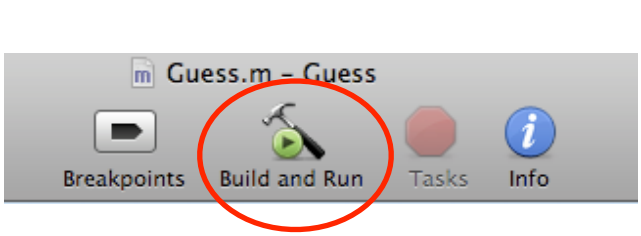
Step9. 開啓Observer.m, 實作這三個method

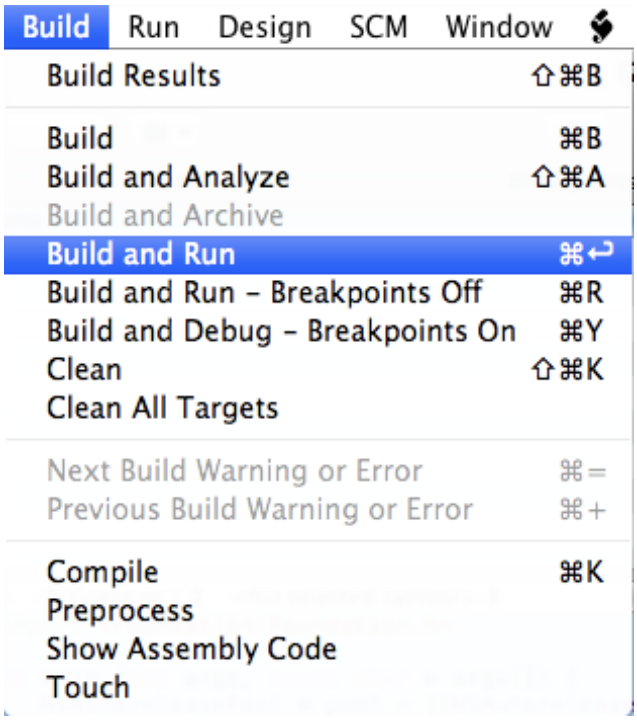
```
#import "Observer.h"


@implementation Observer

- (void) addObserving:(NSObject *) observedObject{
    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(gotNote:) name:@"NameUpdated" object:observedObject];
}

-(void) removeObserving{
    [[NSNotificationCenter defaultCenter] removeObserver:self];
}

-(void) gotNote:(NSNotification *)note{
    NSLog(@"the new name is %@", [[note object] name]);
}

@end
```

Step10. 開啓TestNotification.m, 先 `#import "Engine.h"` 和 `#import "Observer.h"` , 並將印出Hello, World!這行Mark掉, 並加入以下的程式, 主要是新增一個Engine叫做targetEngine, 並持續更改name這個NSString, 但只有第二次更改時才add Observing.

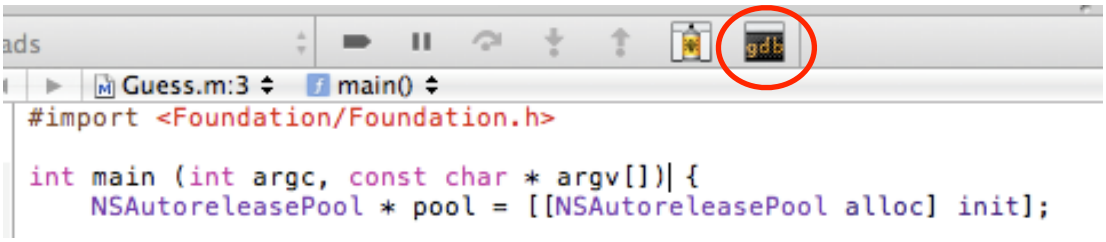
```
#import <Foundation/Foundation.h>
#import "Engine.h"
#import "Observer.h"

int main (int argc, const char * argv[]) {
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

    // insert code here...
    //NSLog(@"Hello, World!");
    Engine * targetEngine = [Engine new];
    Observer * myObserver = [Observer new];
    targetEngine.name = @"initializing";
    [myObserver addObserving:targetEngine];
    targetEngine.name = @"observing";
    [myObserver removeObserving];
    targetEngine.name = @"ending";
    [pool drain];
    return 0;
}
```

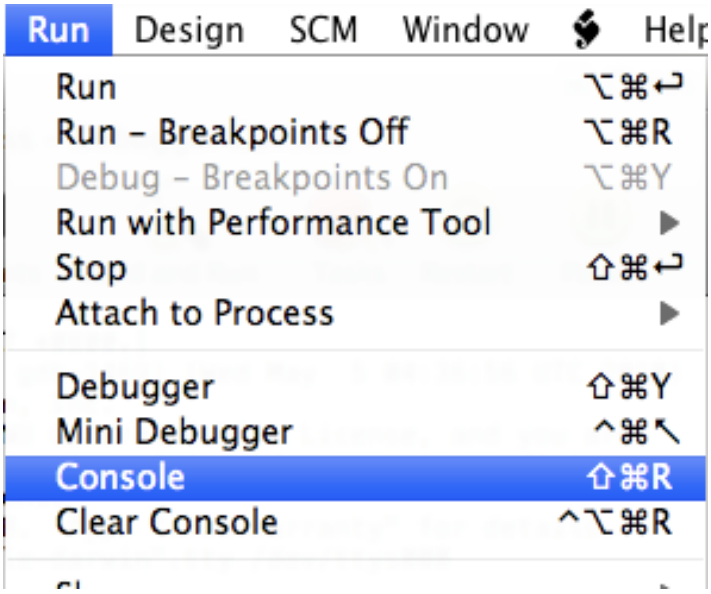
Step11. Build and Run (Command + enter)
在Xcode主頁上按下Build and Run, 或是在Build > Build and Run, 即開始Build code並執行.





可在瀏覽程式視窗上方的 gdb 按下來開啓console, 或是在Run > Console來開啓.





在Console中顯示當name更新為observing時才有被觀察判斷並列印出來