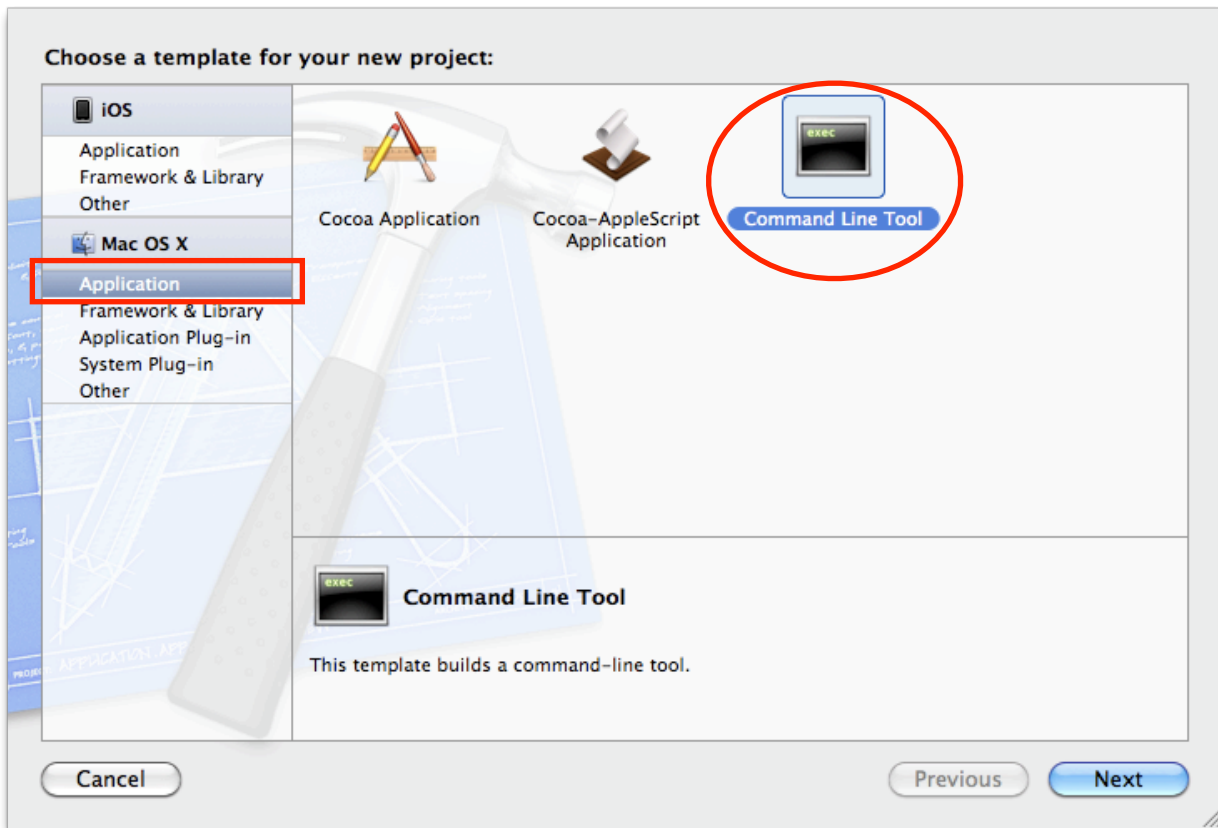
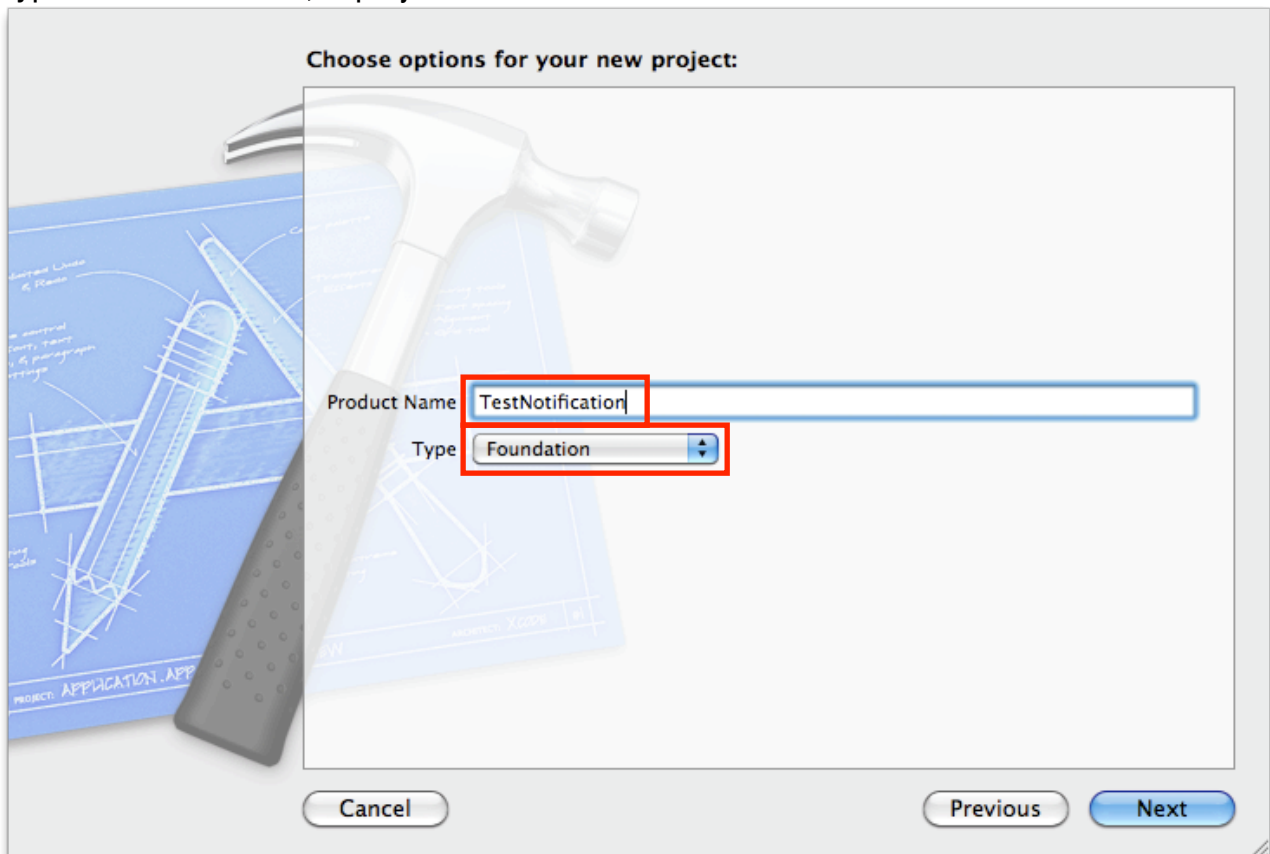


Lab TestNotification

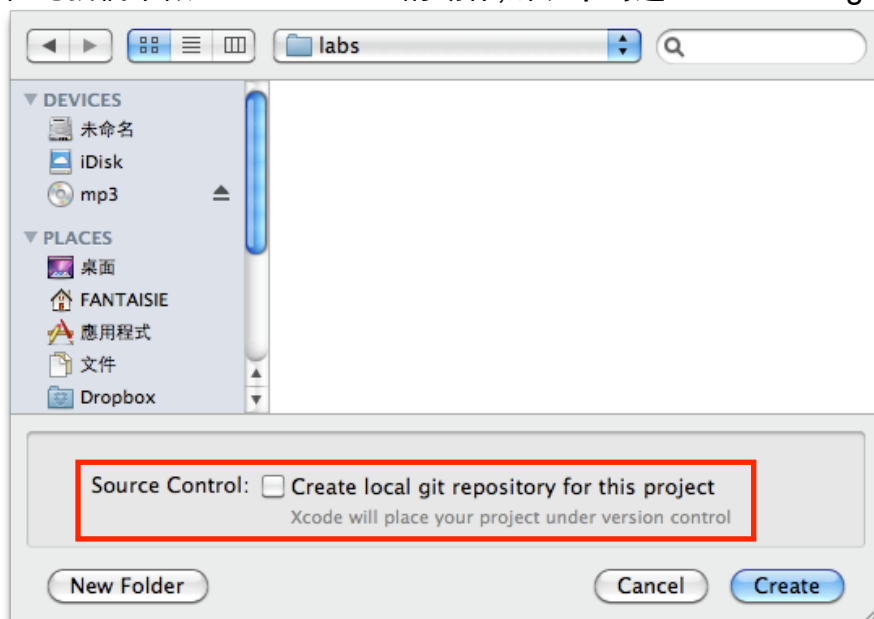
Step 1. 在File > New > New Project開啓一個新的project, 選擇MAC OS X的Application目錄裡的 **Command Line Tool**



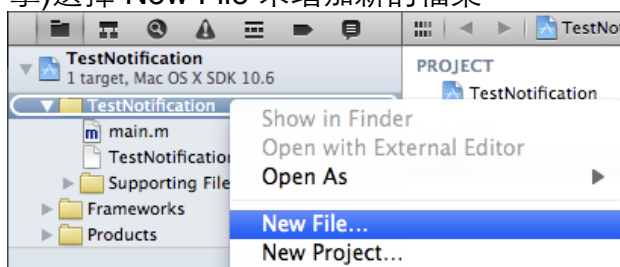
Type選擇**Foundation**, 將project命名為 **TestNotification**



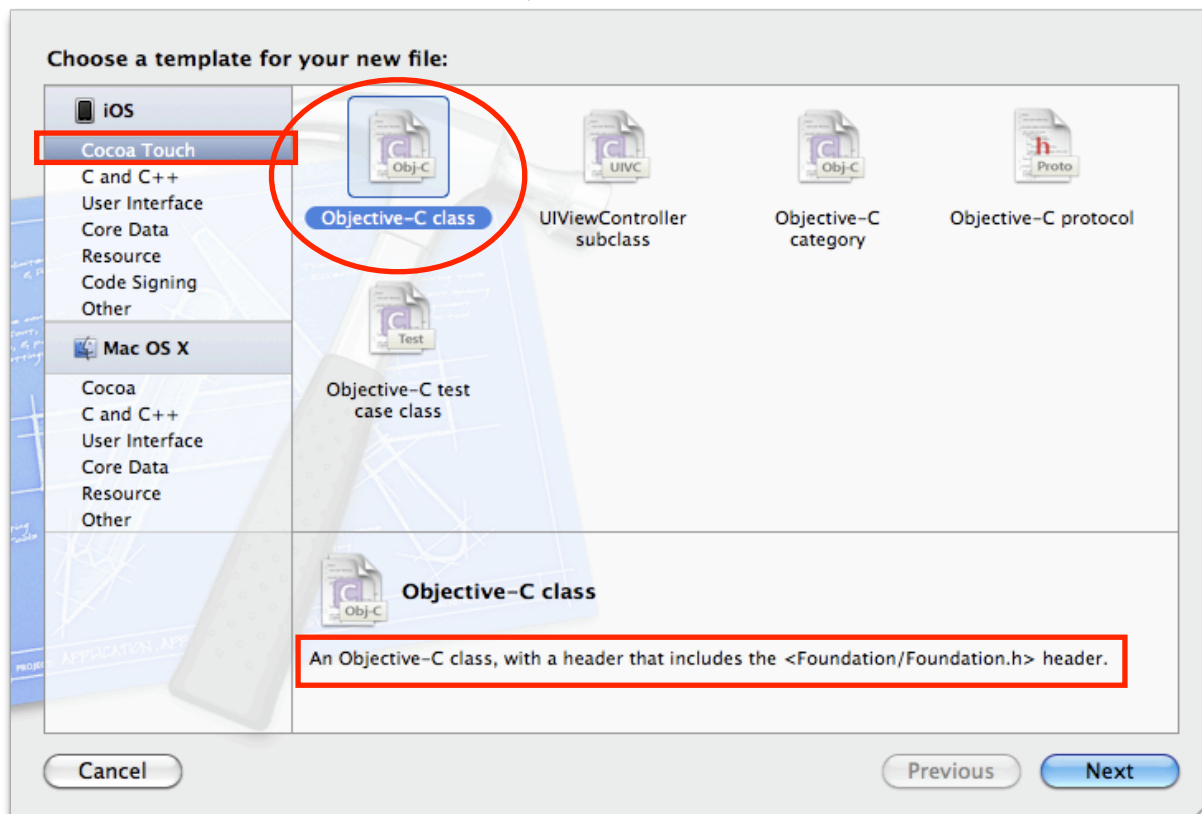
在此我們不做version control的動作,所以不勾選Create local git repository for this project



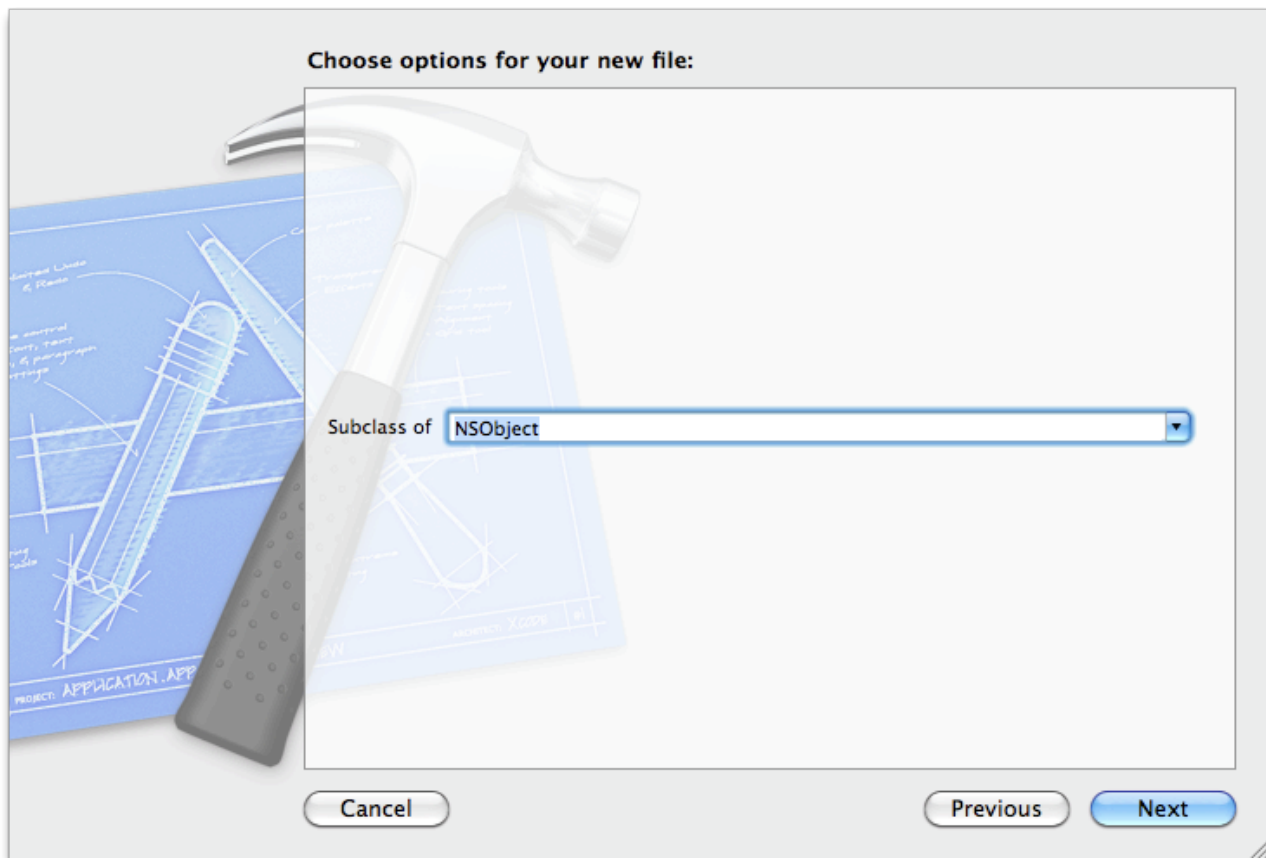
Step 2. 在Xcode左邊Project Navigator 視窗中, 在TestNotification上點右鍵(若無滑鼠ctrl+點擊)選擇 New File 來增加新的檔案



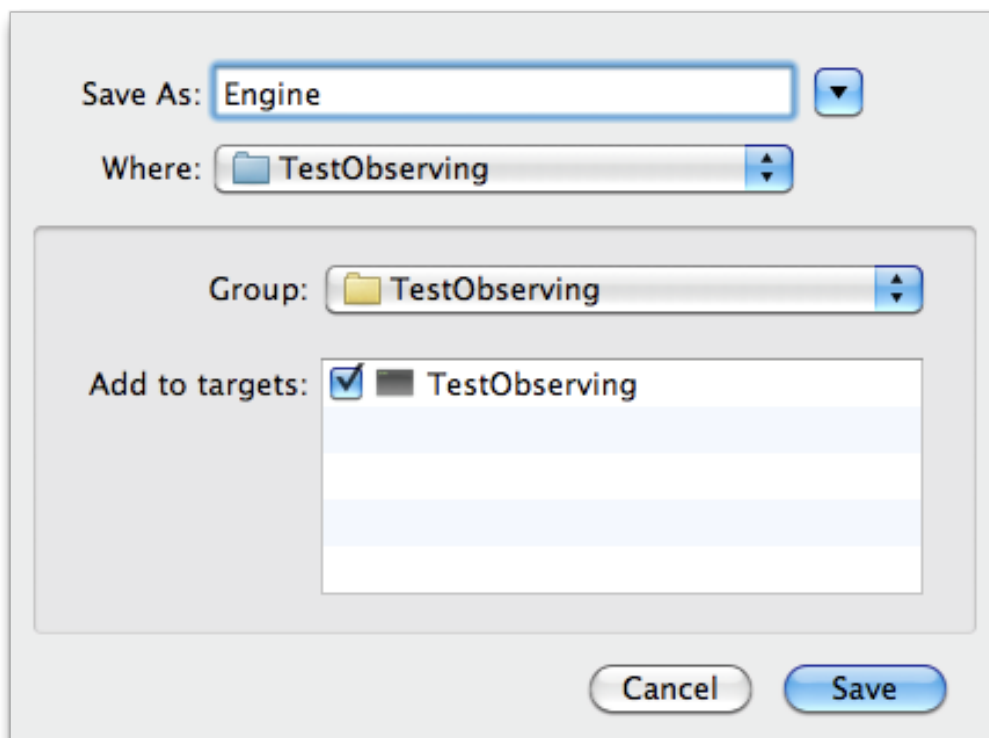
Step 3. 選擇iOS裡的Cocoa Touch 裡的Objective-C class, 下方有敘述這個class有includes <Foundation/Foundation.h> 這個標頭檔



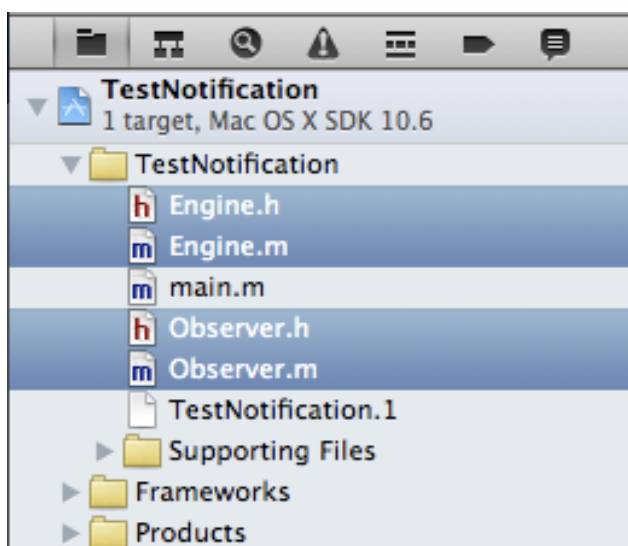
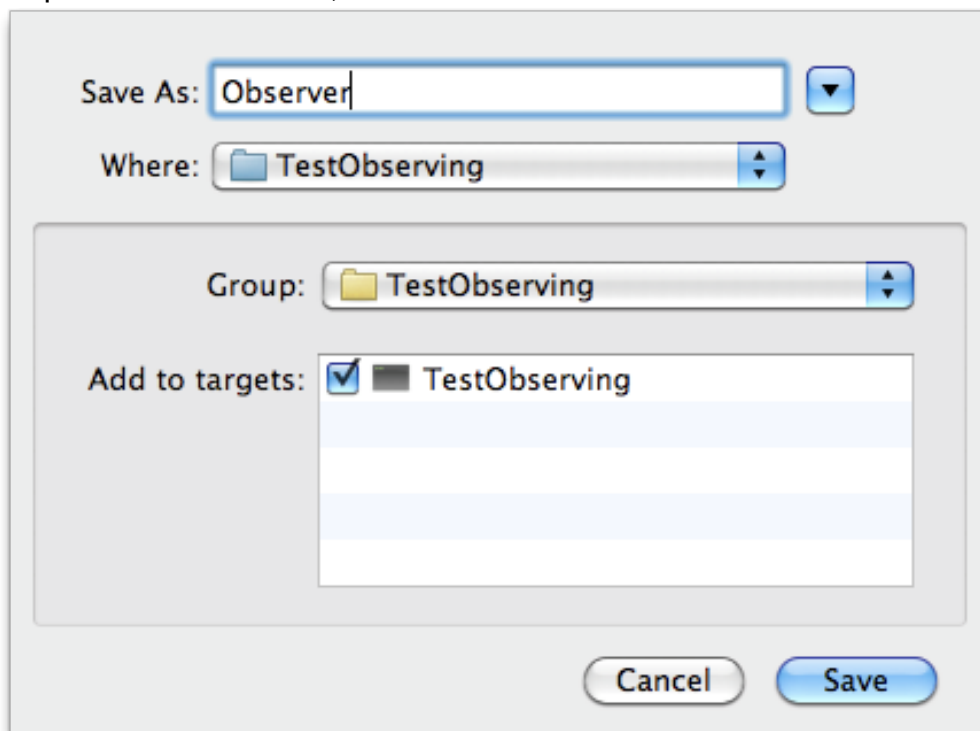
選擇Subclass of NSObject



Step 4. 將新增的Subclass命名為 **Engine** 來產生 Engine.m和 Engine.h



Step 5. 依照上面的方式,再新增一個class叫做 **Observer**



Step 6. 在Xcode左邊 Project Navigator 視窗中,開啓 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
```

Step 7. 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

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

-(NSString *) name{
    return name;
}

@end
```

Step 8. 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) addObserver:(NSObject *) observedObject;
- (void) removeObserver;
- (void) gotNote:(NSNotification *)note;
@end
```

Step9. 開啓Observer.m, 實作這三個method

```
#import "Observer.h"

@implementation Observer

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

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

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

@end
```

Step 10. 開啓**main.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.

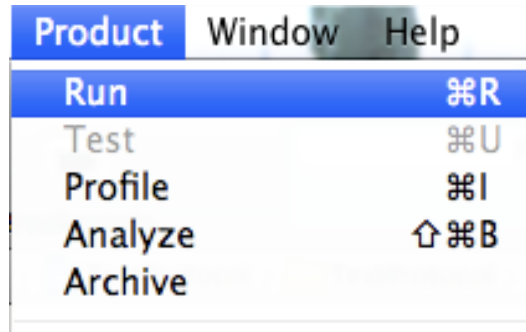
```
#import <Foundation/Foundation.h>
#import "Engine.h"
#import "Observer.h"

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

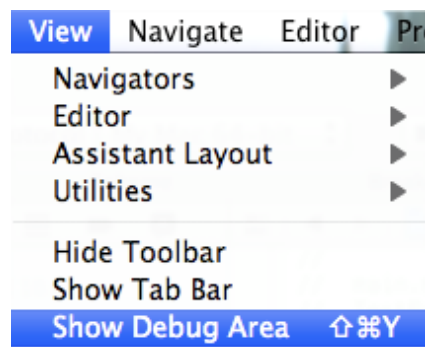
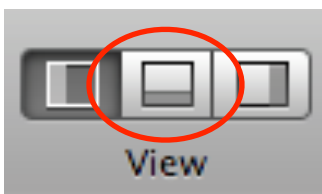
    // insert code here...
    //NSLog(@"Hello, World!");
    Engine * targetEngine = [Engine new];
    Observer * myObserver = [Observer new];
    targetEngine.name = @"initializing";
    [myObserver addObserving:targetEngine];
    targetEngine.name = @"observing";
    [myObserver removeObserving];
   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

Step 11. Run (⌘+R)





在Xcode主頁左上角按下Run, 或是在Product > Run, 即開始Build code並執行



之後會自動開啓console, 沒開啓的話在右上角View點擊中間的Button, 或是選擇View > Show Debug Area 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

```
All Output  Clear     
GNU gdb 6.3.50-20050815 (Apple version gdb-1518) (Sat Feb 12 02:52:12 UTC 2011)  
Copyright 2004 Free Software Foundation, Inc.  
GDB is free software, covered by the GNU General Public License, and you are  
welcome to change it and/or distribute copies of it under certain conditions.  
Type "show copying" to see the conditions.  
There is absolutely no warranty for GDB.  Type "show warranty" for details.  
This GDB was configured as "x86_64-apple-darwin".tty /dev/ttys000  
[Switching to process 2097 thread 0x0]  
2011-04-02 21:31:27.850 TestNotification[2097:903] the new name is observing  
Program ended with exit code: 0
```