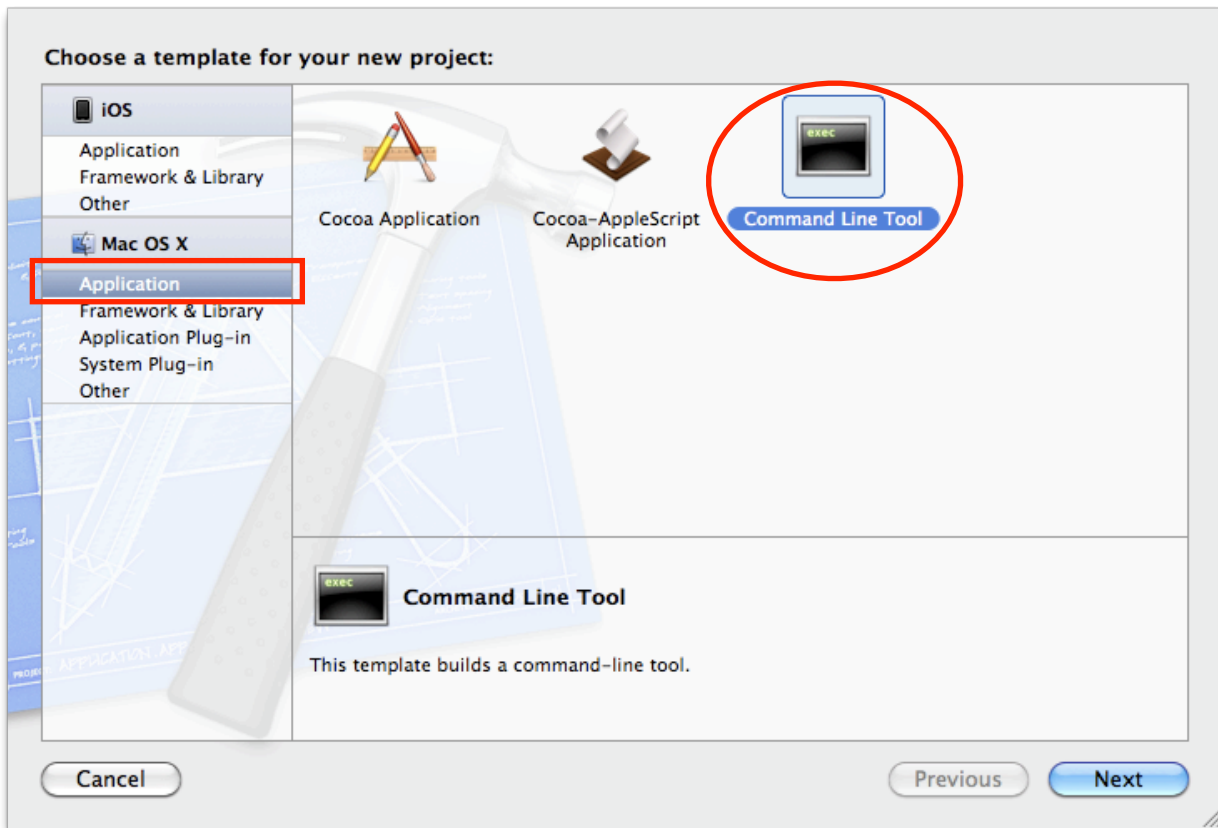
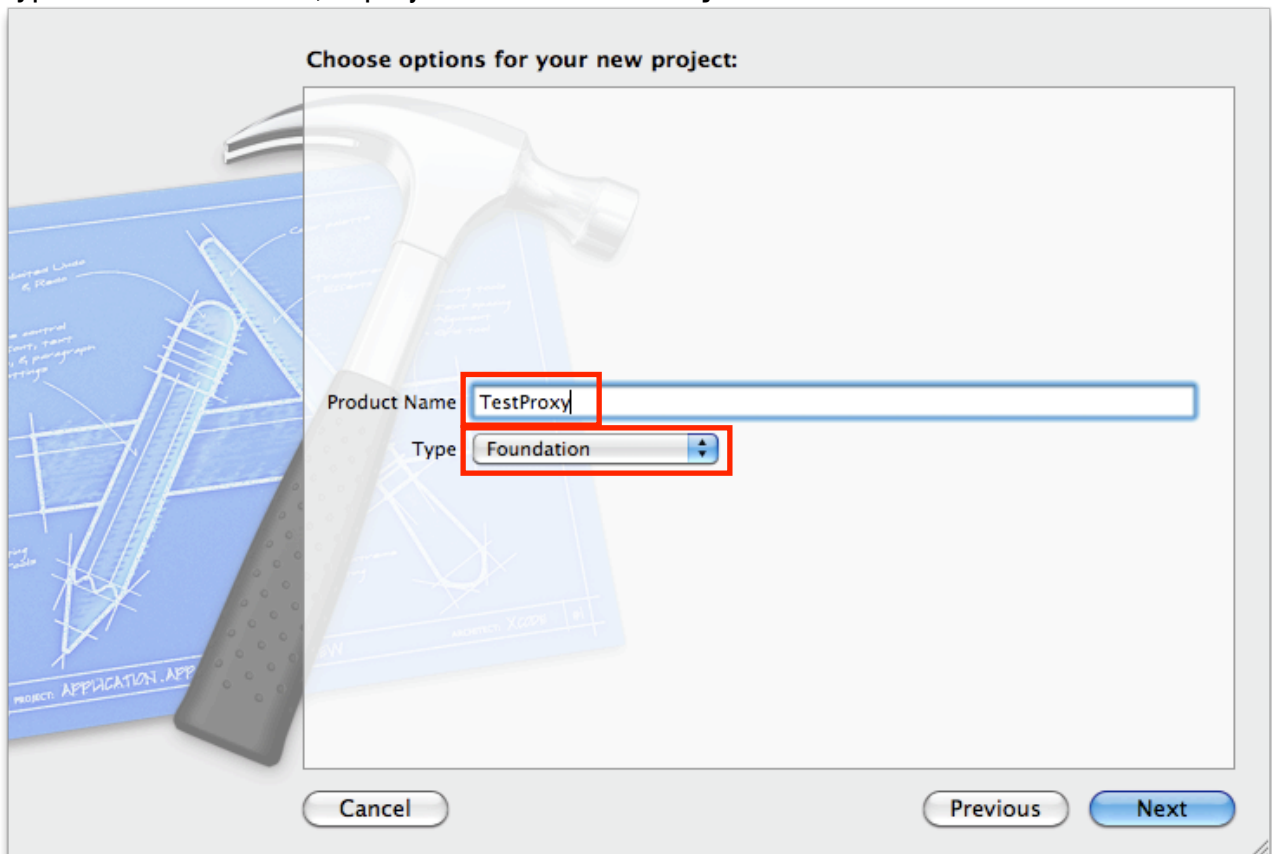


Lab TestProxy

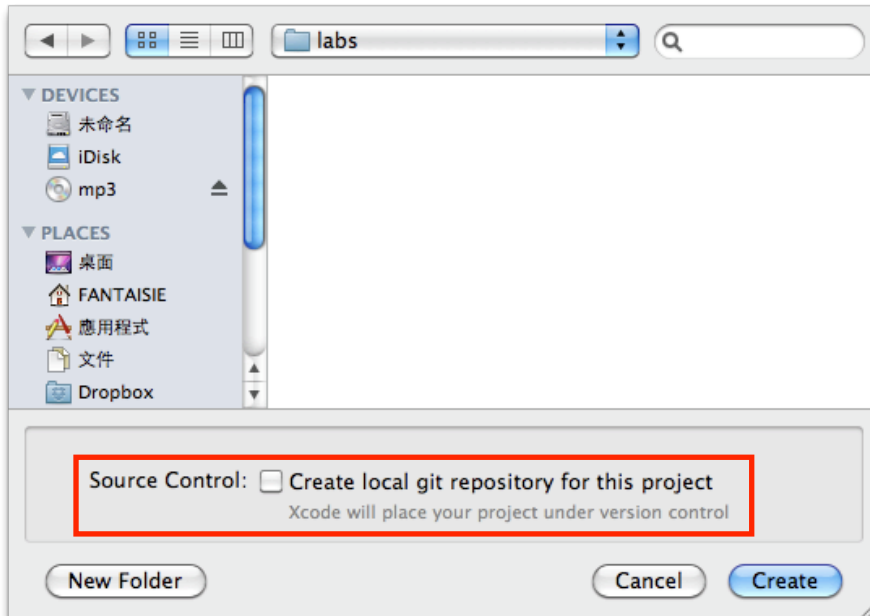
Step 1. 在File > New > New Project開啓一個新的project, 選擇MAC OS X的Application目錄裡的 **Command Line Tool**



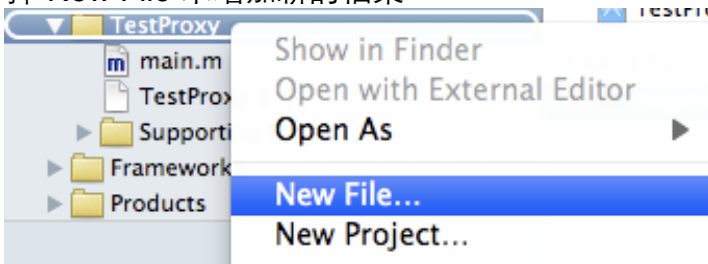
Type選擇 **Foundation**, 將project命名為 **TestProxy**



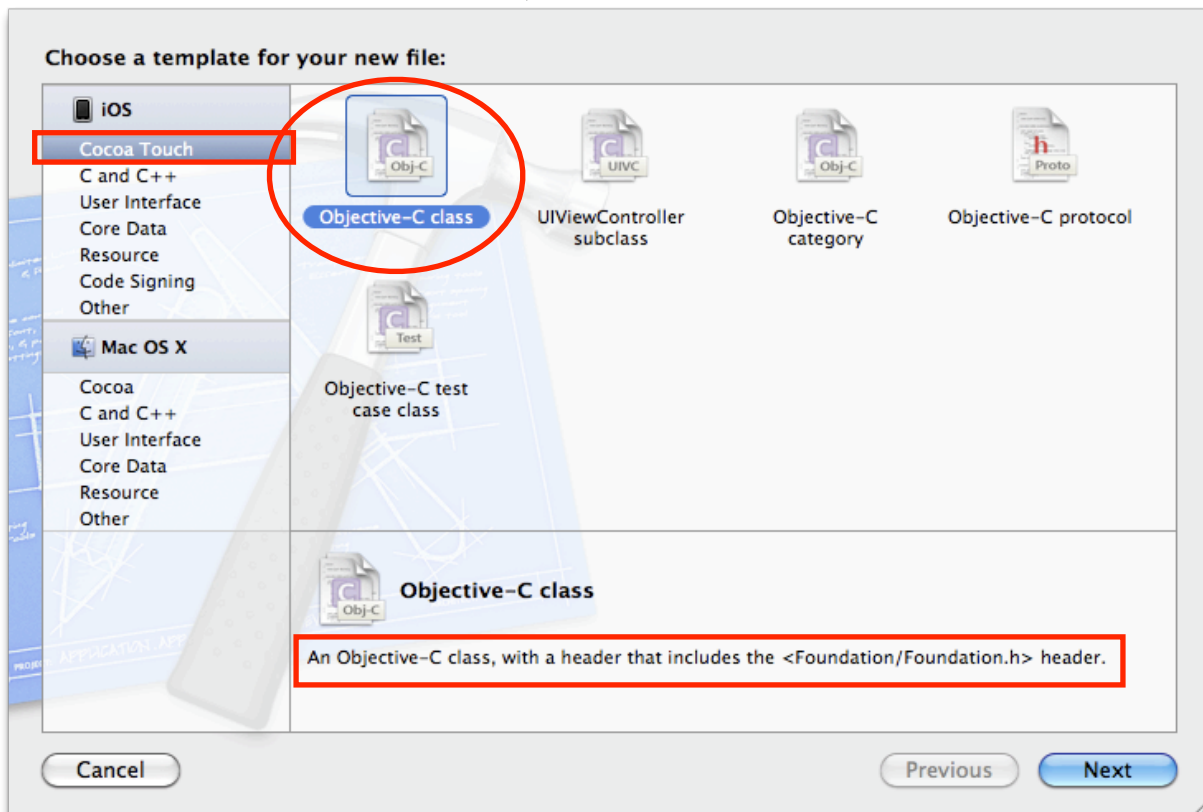
在此我們不做version control的動作,所以不勾選Create local git repository for this project



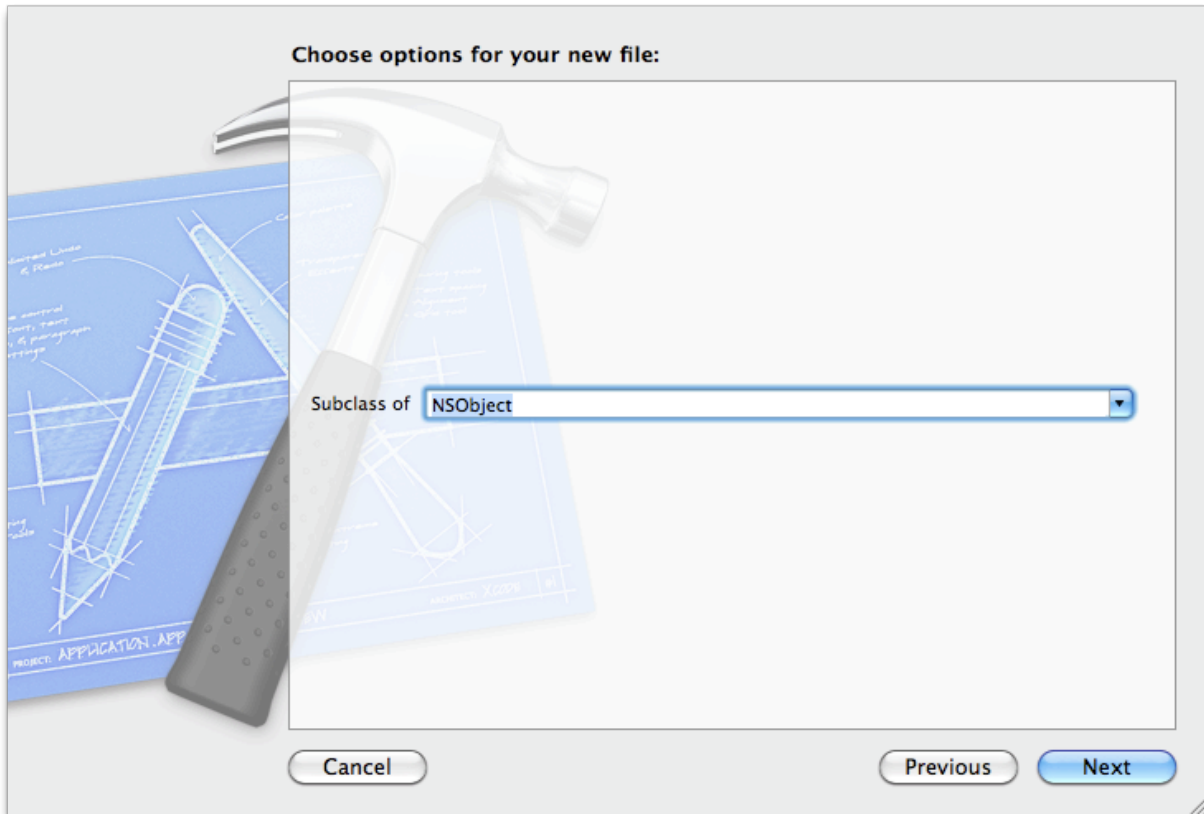
Step 2. 在Xcode左邊Project Navigator 視窗中, 在TestTimer上點右鍵(若無滑鼠ctrl+點擊)選擇 New File 來增加新的檔案



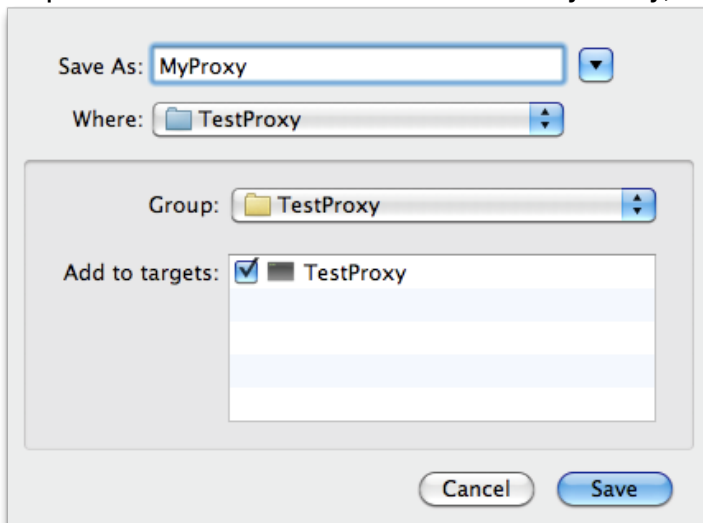
Step 3. 選擇iOS裡的Cocoa Touch 裡的Objective-C class, 下方有敘述這個class有includes <Foundation/Foundation.h> 這個標頭檔



選擇Subclass of NSObject



Step 4. 將新增的subclass的File命名為MyProxy, 來產生 MyProxy.m 和 MyProxy.h



Step 5. 開啓MyProxy.h, 先將原來繼承的 NSObject 改成 NSProxy, 宣告我們會代理的Object (因無法知道為什麼Class, 設為id), 並設定property來宣告accessor.

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

@interface MyProxy : NSProxy {
    id targetObject;
}

@property (retain, nonatomic) id targetObject;

@end
```

Step 6. 開啓MyProxy.m, 除了實作建構子 init 之外(必需), 還必須實作覆寫兩個 NSProxy 的 method: forwardInvocation: 和 methodSignatureForSelector:

```
#import "MyProxy.h"

@implementation MyProxy
@synthesize targetObject;

-(id) init{
    return self;
}

-(void)forwardInvocation:(NSInvocation *)anInvocation {
    [anInvocation setTarget:targetObject];
    [anInvocation invoke];
}

-(NSMethodSignature *)methodSignatureForSelector:(SEL)aSelector {
    return [targetObject methodSignatureForSelector:aSelector];
}

@end
```

Step 7. 開啓main.m, 先 import "MyProxy.h" , 然後將印出Hello, World!的這行去掉, 新增一個NSMutableArray的物件newArray, 以及一個MyProxy的物件newProxy, 在Array放進一個NSString的物件之後, 接下來就把工作交由newProxy代理, 然後由newProxy來新增另外一個NSString的物件, 最後透過NSLog列印出來.

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

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

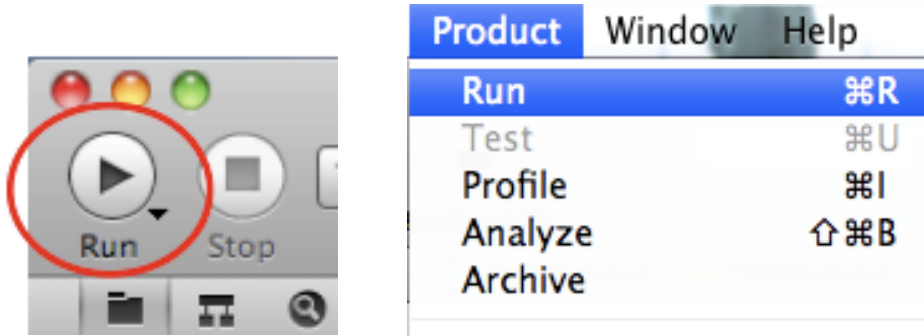
    // insert code here...
    //NSLog(@"Hello, World!");

    NSMutableArray *newArray = [NSMutableArray arrayWithObject:@"Hello
from Me"];
    MyProxy *newProxy = [[MyProxy alloc] init];
    newProxy.targetObject = newArray;
    [newProxy addObject:@"Hello from Proxy"];
    NSLog(@"%@", newProxy.targetObject);

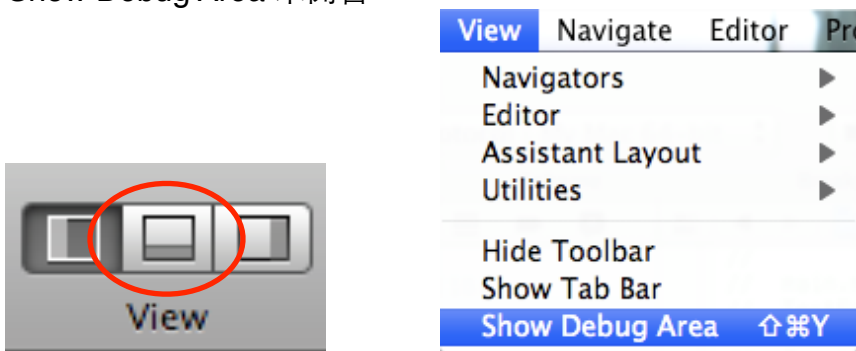
    [pool drain];
    return 0;
}
```

Step 8. 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



印出由newArray自己放進的NSString和由newProxy新增的一個NSString

```
All Output ↑ [Clear] [Icons]
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/ttys002
[Switching to process 5119 thread 0x0]
2011-04-03 20:31:12.344 TestProxy[5119:903] (
    "Hello from Me",
    "Hello from Proxy"
)
Program ended with exit code: 0
```

但在 main.m 會顯示 **"MyProxy" may not respond to "-addObject:"**

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

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

    // insert code here...
    NSLog(@"Hello, World!");
    NSMutableArray *newArray = [NSMutableArray arrayWithObject:@"Hello from Me"];
    MyProxy *newProxy = [[MyProxy alloc] init];
    newProxy.targetObject = newArray;
    [newProxy addObject:@"Hello from Proxy"];
    NSLog(@"%@", newProxy.targetObject);
    [pool drain];
    return 0;
}
```

⚠️ "MyProxy" may not respond to 'addObject:'