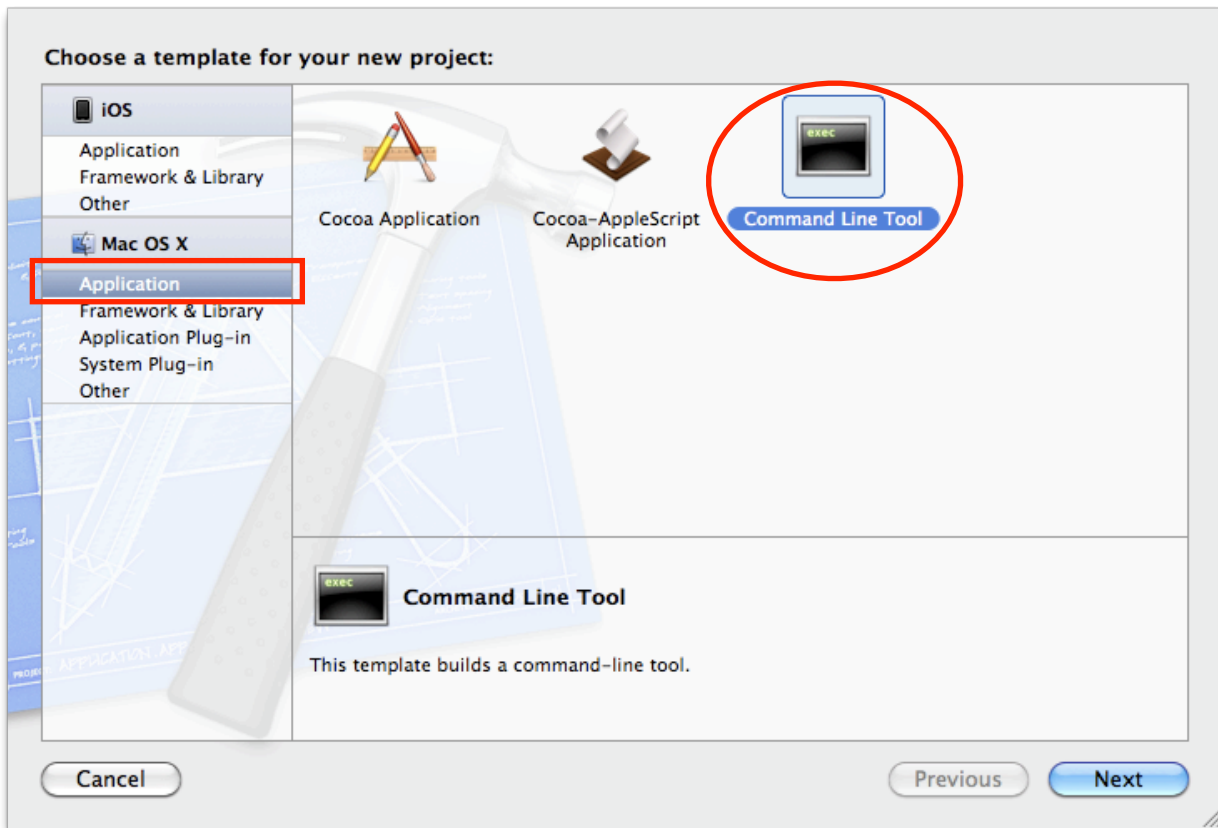
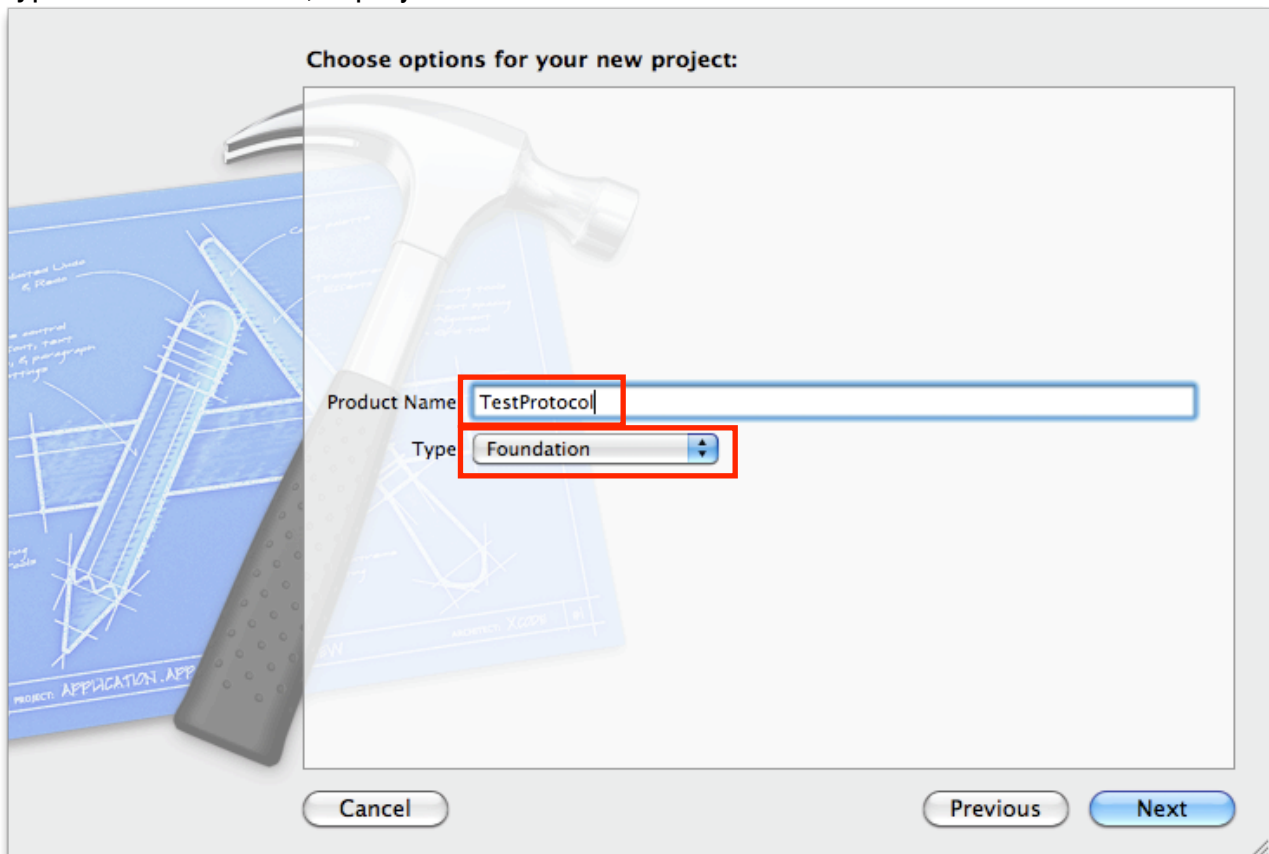


Lab TestProtocol

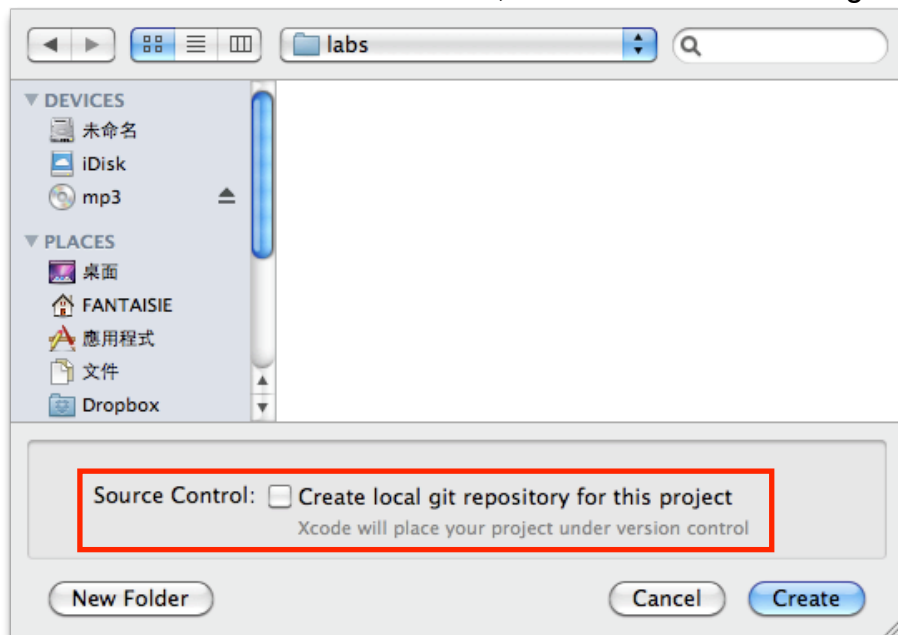
Step 1. 在File > New > New Project開啓一個新的project, 選擇MAC OS X的Application目錄裡的 **Command Line Tool**



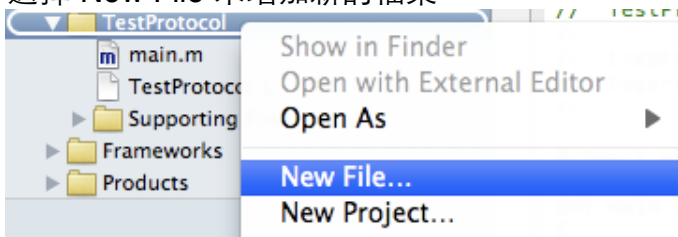
Type選擇**Foundation**, 將project命名為 **TestProtocol**



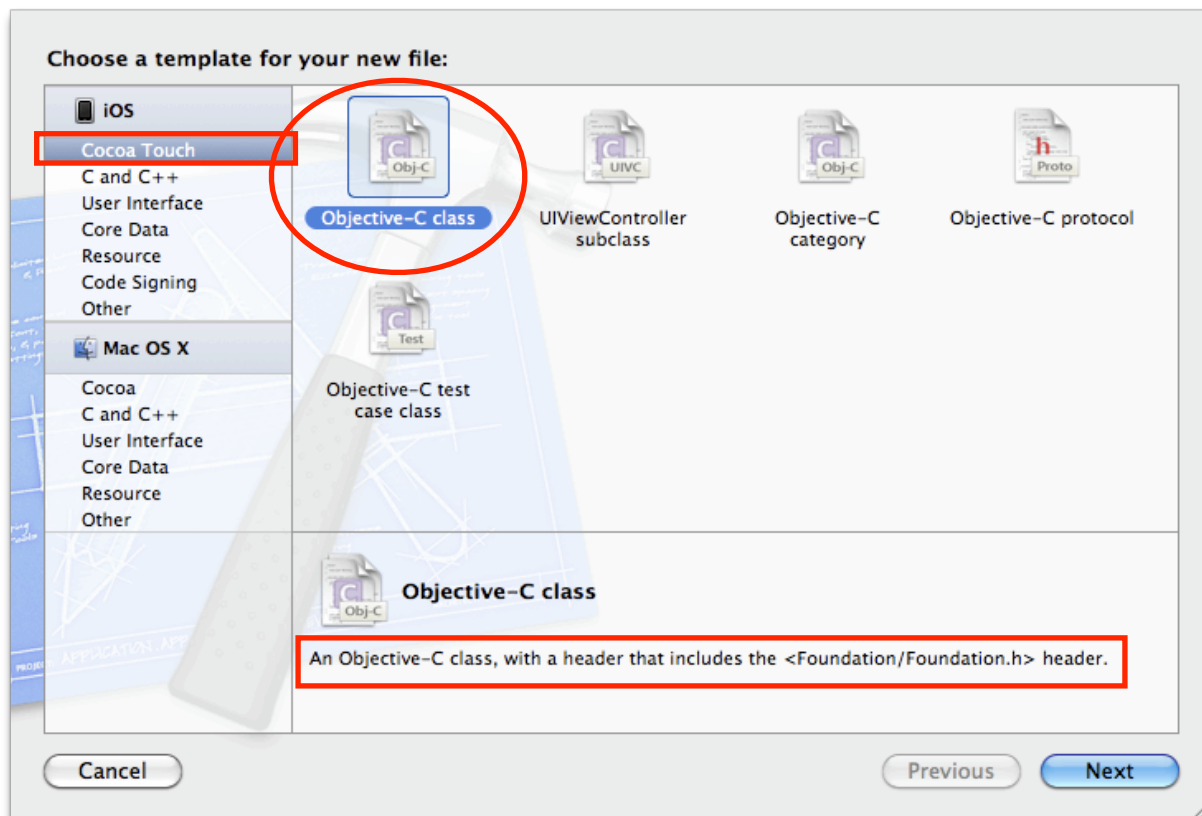
在此我們不做version control的動作,所以不勾選Create local git repository for this project



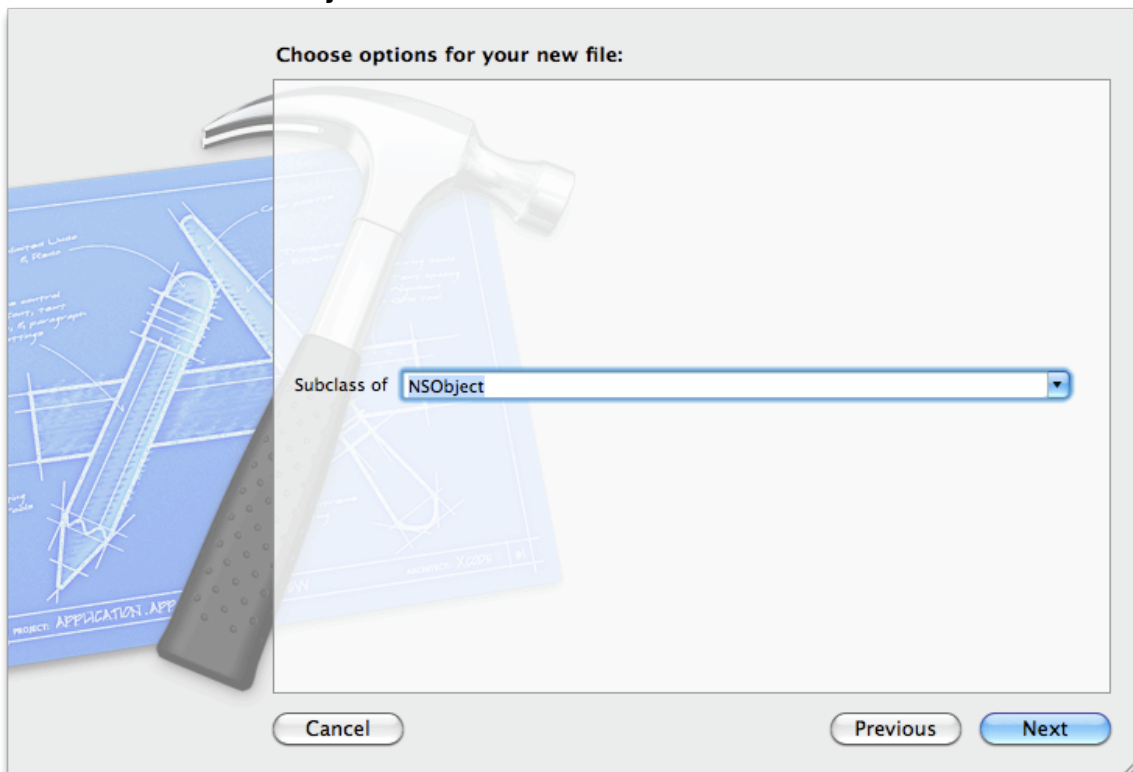
Step 2. 在Xcode左邊Project Navigator 視窗中, 在TestProtocol上點右鍵(若無滑鼠ctrl+點擊) 選擇 New File 來增加新的檔案



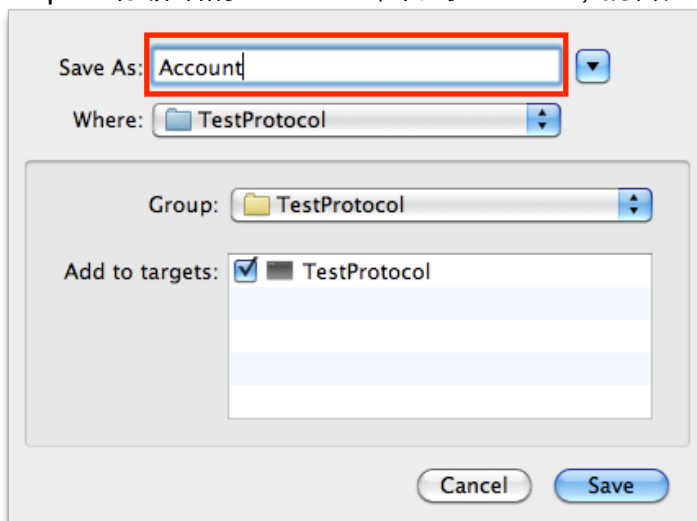
Step 3. 選擇iOS裡的Cocoa Touch 裡的Objective-C class, 下方有敘述這個class有includes <Foundation/Foundation.h> 這個標頭檔



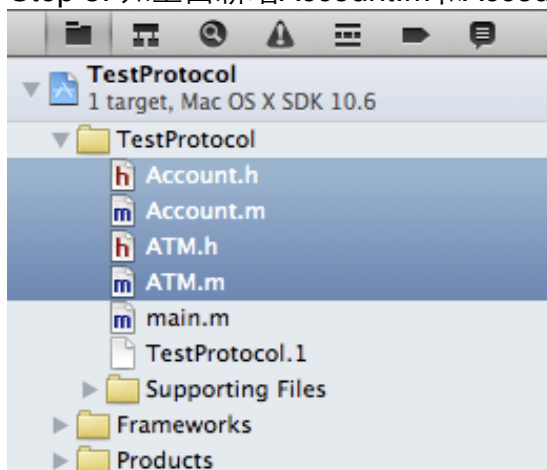
選擇Subclass of NSObject



Step 4. 將新增的subclass命名為 Account, 就會產生 Account.m 和 Account.h



Step 5. 如上面新增Account.m和Account.h一樣的步驟來新增ATM.m和ATM.h



Step 6. 開啓 Account.h, 在裡面宣告withdraw這個protocol和他的method,之後會交由ATM這個NSObject作為Account的delegate去實作. 並將幾個變數的property都設定好, 並加入一個method叫做takeMoney:money:

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

```
@protocol withdraw;
```

```
@interface Account : NSObject {
    NSString *name;
    int deposit;
    id<withdraw> delegate;
}
-(void) takeMoney:(int) money;
@property (retain) NSString * name;
@property (assign) int deposit;
@property (assign) id<withdraw> delegate;
@end
```

```
@protocol withdraw<NSObject>
```

```
-(void) withdrawMoney:(Account *) account withName:(NSString *) name
withMoney:(int) money;
```

```
@end
```

Step 7. 開啓Account.m,將剛剛已設定property的三個變數做synthesize, 並且實作takeMoney:money這個method, 主要就是如果Account的delegate有實作(respondToSelector) withdrawMoney:withName:withMoney: 這個method的話,就把withdrawMoney這件工作交由delegate去完成.

```
#import "Account.h"
```

```
@implementation Account
```

```
@synthesize name, deposit, delegate;
```

```
-(void) takeMoney:(int)money{
    if ([delegate respondsToSelector:@selector
(withdrawMoney:withName:withMoney:)])
        [delegate withdrawMoney:self withName:name withMoney:money];
    else
        NSLog(@"Please implement withdrawMoney:withName:withMoney:");
}
@end
```

Step 8. 開啓ATM.h並在裡面 #import "Account.h" 讓 ATM 知道 <withdraw> 和 Account 的存在,在 Class 最後寫 NSObject<withdraw> 來繼承 NSObject 而且遵循 <withdraw> 的規範

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

```
#import "Account.h"
```

```
@interface ATM : NSObject<withdraw> {
```

```
}
```

```
@end
```

Step 9. 開啓ATM.m,並在裡面實作 `withdrawMoney:withName:withMoney:` 這個method, 這個method的主要工作就是將領出的money從deposit扣掉,並將相關的資訊列印出來.

```
#import "ATM.h"
```

```
@implementation ATM
```

```
-(void) withdrawMoney:(Account *) account withName:(NSString *) name
withMoney:(int) money{

    if(account.deposit >= money){
        account.deposit -= money;
        NSLog(@"***Name: %@, Withdraw: %d, Deposit: %d***",
account.name, money, account.deposit);
    }
    else {
        NSLog(@"***Not enough money!! - Name: %@, Deposit:
%d***",account.name, account.deposit);
    }
}

}
```

```
@end
```

Step 10. 開啓main.m, 先 `#import "Account.h"` 和 `#import "ATM.h"` ,並將印出Hello, World!這行Mark掉, 並加入以下的程式, 來建立一個account並設定name和deposit,之後再透過atm作為delegat去扣除領出的錢並將資訊列印出來

```
#import <Foundation/Foundation.h>
#import "Account.h"
#import "ATM.h"
```

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

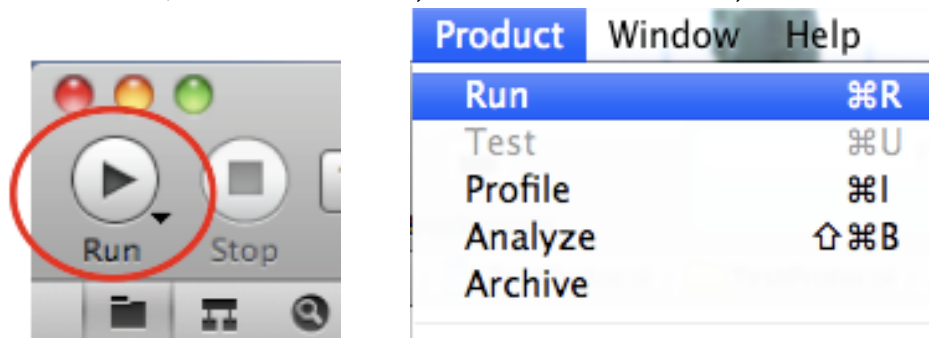
    // insert code here...
    //NSLog(@"Hello, World!");
```

```
Account *account = [Account new];
account.name = @"Michael";
account.deposit = 1000;
ATM *atm = [ATM new];
account.delegate = atm;
[account takeMoney:700];
[account takeMoney:500];
```

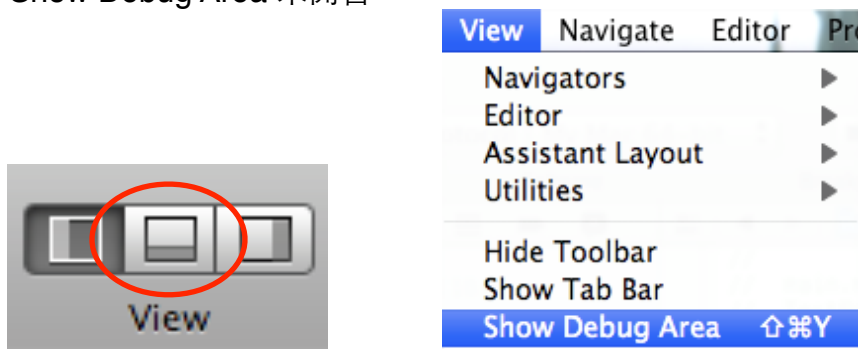
```
[pool drain];
return 0;
}
```

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中顯示第一次withdraw了700, 還剩下300, 而第二次要再領500時, money已經餘額不足了

```
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 1221 thread 0x0]
2011-03-31 23:11:46.025 TestProtocol[1221:903] ***Name: Michael, Withdraw: 700, Deposit: 300***
2011-03-31 23:11:46.028 TestProtocol[1221:903] ***Not enough money!! - Name: Michael, Deposit:
300***
Program ended with exit code: 0
```

//