Week 1 Practicum assignment

**Hellglow World**



Introduction  
In this first assignment we will take a look at the basics of Objective C.

To do this properly, the first project will not be an iPhone project.

Further along you’ll get the chance to make that first iOS app!.

Another important part is learning to work with SVN.

# Part 1: Getting SVN

You won’t need SVN until part X of this assignment, but it takes a while for the repository to be created.

So go to the following address to get your SVN repo:.

<https://portal.fhict.nl/Studentenplein/Lists/Aanvraag%20SVN%20Repository/newform.aspx>

Don’t forget to also allow your teachers access to the repository!

Homework: Read the SVN manual at:

[https://portal.fhict.nl/ISM/p5p62012/Shared%20Documents/Version%20control%20with%20SVN.docx](https://portal.fhict.nl/ISM/p5p6/Shared%20Documents/Version%20control%20with%20SVN.docx)

There might be a pop quiz next week!

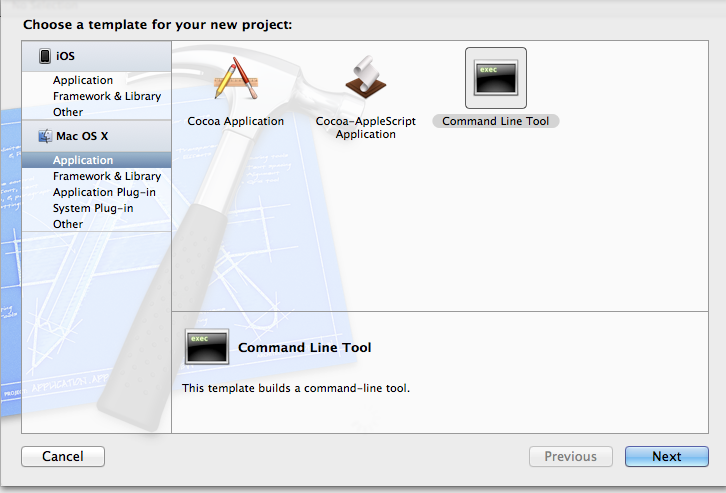
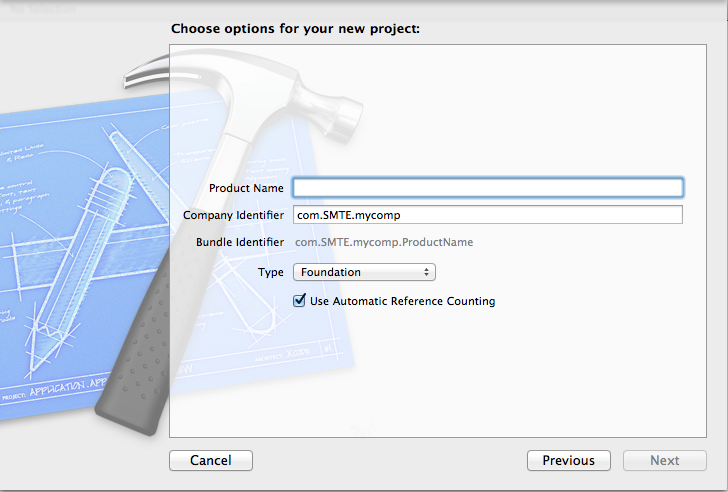
# Part 2: HellGlow world

## Step 1 The start

It is custom for the first assignment to somehow display Hello World.

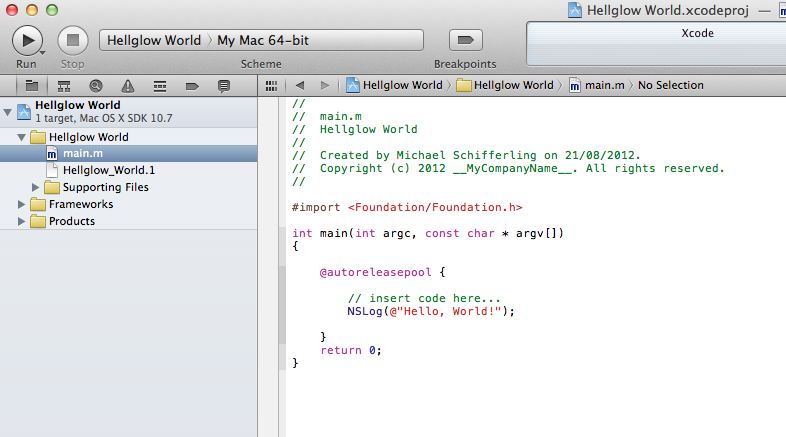
So we do something like that as well.

Start the XCode IDE, and create a new project. Select the Mac OSX application and choose ‘Command Line Tool’

Give the project a good name(HellGlow World) and company identifier (com.yourname) and save it. Check the ‘Use automatic Reference Counting’ checkbox. In the next dialog Uncheck the ‘Create Local GIT repository checkbox’

When done right you get something like this:



## Step 2 Glowing

Right now it’s almost ready for running, but the text is still wrong.

Change the line



to

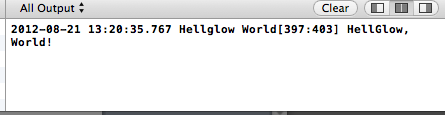


Done.

Now it’s time to run the application by clicking ‘Run’



Take a good look at the console output, you should see something like this:



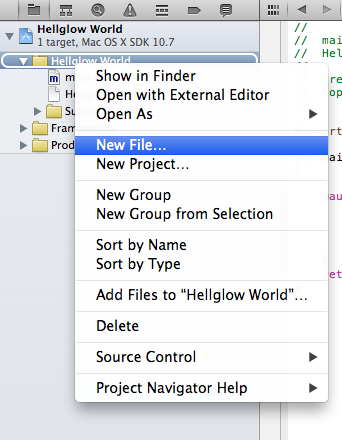
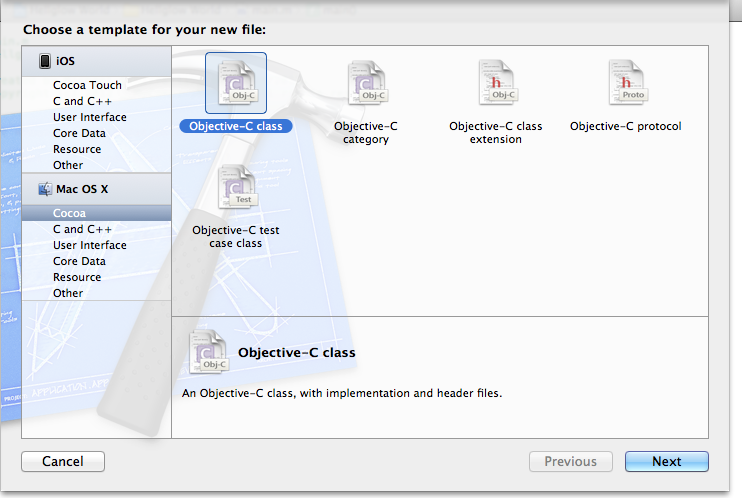
## Step 3 Making it classy

Time to spice it up a bit.

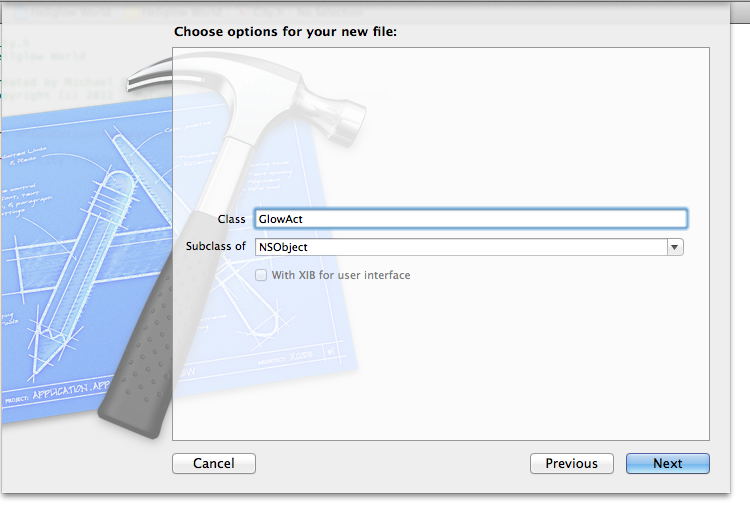
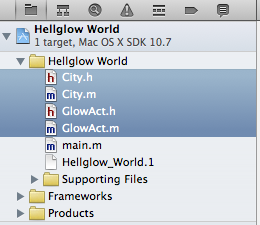
Glow is all about doing cool stuff with Lights in Eindhoven.

Let’s create two classes: GlowAct and City.

XCode makes this easy for you, just right click on the folder in the organizer and choose ‘new file’. In the following dialog select ‘Objective C class’ template.

In the next dialog give the class the name and choose the subclass NSObject

Done.

Perform the same steps for the City class.

## Step 4 public properties

Xcode was kind enough to generate the .m and .h files for you, but they are both pretty useless. It’s up to you to add some extra functionality to the classes.

Add the following properties to the class GlowAct.

(Don’t forget to synthesize them as well)

* NSString name
* NSInteger rating
* NSString startTime

Also add the following properties to the City class

* NSString name
* NSInteger population
* NSMutableArray glowActs

## Step 5 Showing off

At this point the classes can store data in the properties, but they don’t actually do something with it. Let’s start with a showInfo method, so the classes can tell what data is set.

Add a method to both classes called showInfo. It should return nothing, and take no parameters.

In the implementation of the GlowAct showInfo method add the following:



NSLog is used to display data and the %@ thingies are called string formatters.

Don’t worry about it for now, we will cover this next week.

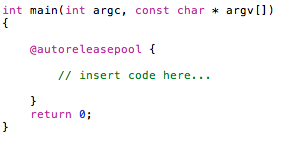
In the implementation of the City showInfo method add the following



## Step 6 Classy objects

Ok. Let’s do something with these classes.

Open the main.m file, and adjust the main method below the //Enter code here.. line.



Create a new object from the GlowAct class. Call it blueLightAct.

Set the properties with the following values.

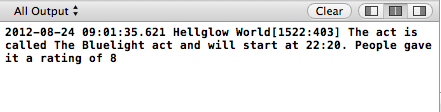
Name: The Bluelight act.

Start time: 22:20

Rating: 8

Finally call the showInfo method.

The output should look like this:



## Step 7 Classy objects 2

Now you also need an object of the type City.

Create it after the GlowAct, and give the properties these values:

Name: Eindhoven

Population: 220000

Finally also call it’s showInfo method.

## Step 8 Init the init

You might have noticed that the City class also has an NSMutableArray that is not currently used. The reason for that is, that the NSMutableArray has to be created first. Add an init method to the City class where the NSMutableArray is created. After that don’t forget to actually use the init.

When done right you should now be able to add the blueLightAct object to the array by calling the addObject method of the glowActs array.

## Step 9 Really showing off

When you call the showInfo method of the city, the number of glowActs is not displayed. Adjust this so thenumber of glowActs in the NSMutableArray are shown.

**Bonus**: Create another glowAct and add it to the city. Further adjust the showInfo method to use a for loop to show all the act’s info as well.

# Part 3 HellGlow world iOS

The fine people were kind enough to make a very thorough Hello World on iOS tutorial.

You can find it here.

<http://developer.apple.com/library/ios/#DOCUMENTATION/iPhone/Conceptual/iPhone101/Articles/01_CreatingProject.html>

Bonus points if you manage to make it display HellGlow world.

# Part 4: Get it in the Repo

Since you’ve finished two projects, it’s time to put them in the SVN repository.

Check out the repository on your machine.

Create a new directory in your local repository called ‘week1’ and commit it to the repository.

Finally copy the Hello World projects to the week 1 directory and add them the you local repository. Finally commit them with a good commit message.