

Object-oriented Programming

When you make lots of one thing.
It's one thing.

But you can make a bunch of them.

Why OOP?

- Attributes are linked (contrast arrays)
- Create many of one type of object
- Organize your code; avoid driving yourself mad

Dot notation

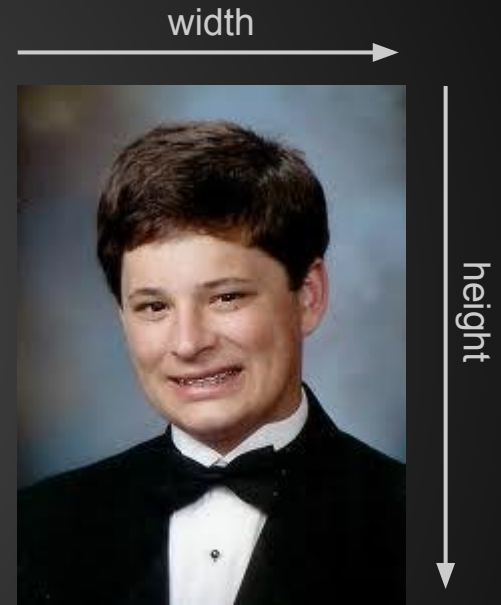
This is how we access the attributes of an object.

You've seen this before:

ofImage (or PImage in Processing)

```
ofImage yearbookPicture;
```

```
yearbookPicture.height;  
yearbookPicture.width;
```

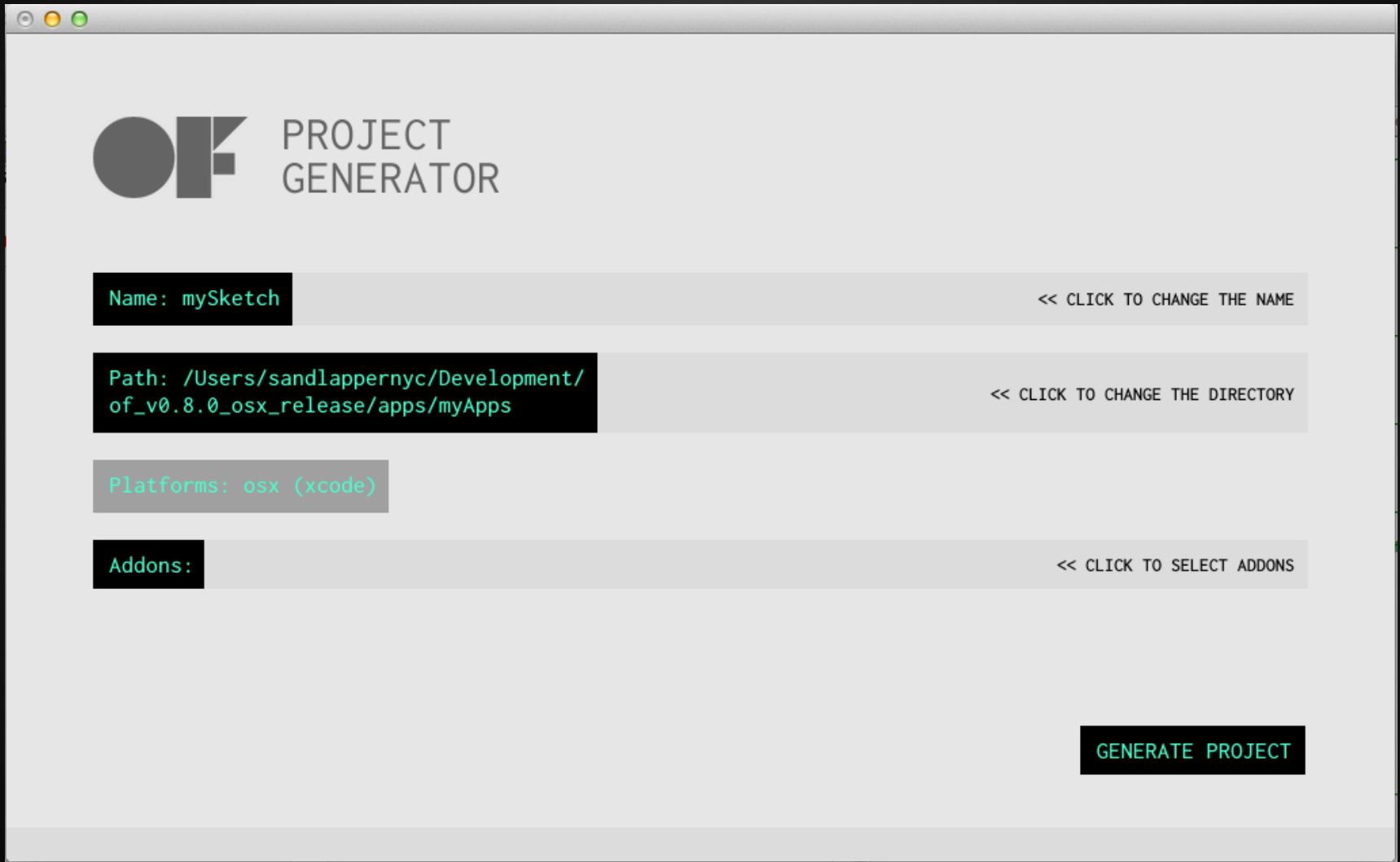


Width and length are variables within the ofImage class

And so it is with all objects...

enemyOne.health;	enemyOne.armorColor;
enemyTwo.health;	enemyTwo.armorColor;

Project generator



The screenshot shows a web application titled "PROJECT GENERATOR" with a logo consisting of a circle and a stylized 'F'. The interface includes four input fields for project configuration, each with a "CLICK TO" action link. The "Name" field contains "mySketch", the "Path" field contains a long directory path, the "Platforms" field contains "osx (xcode)", and the "Addons" field is empty. A "GENERATE PROJECT" button is located at the bottom right.

PROJECT GENERATOR

Name: mySketch << CLICK TO CHANGE THE NAME

Path: /Users/sandlappernyc/Development/of_v0.8.0_osx_release/apps/myApps << CLICK TO CHANGE THE DIRECTORY

Platforms: osx (xcode)

Addons: << CLICK TO SELECT ADDONS

GENERATE PROJECT

How to create a class in oF

Like testApp.h and testApp.cpp, classes in oF (and C++) are typically contained in two separate files.

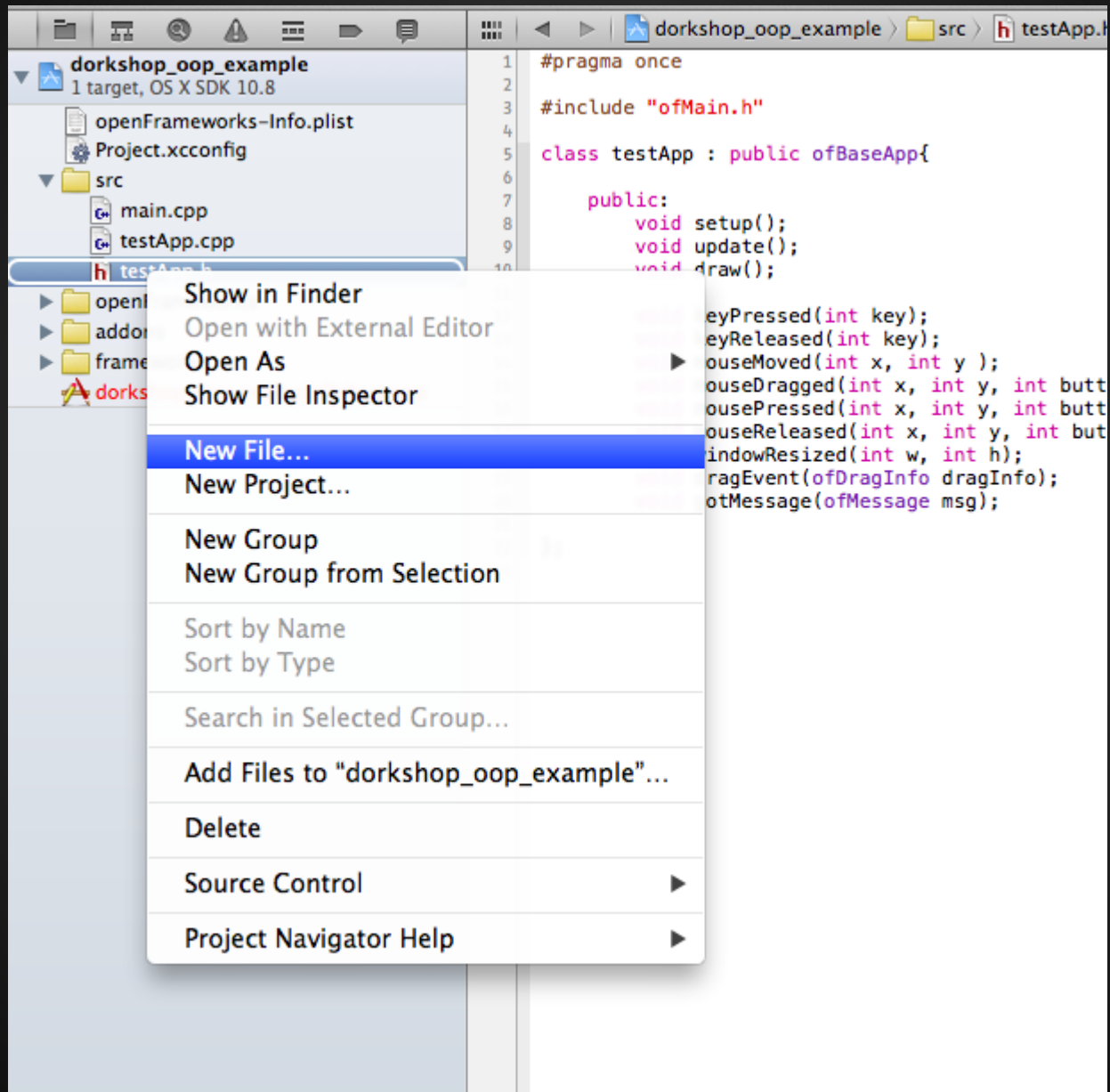
Header file: “AwesomeClass.h”

- Lists variables and methods

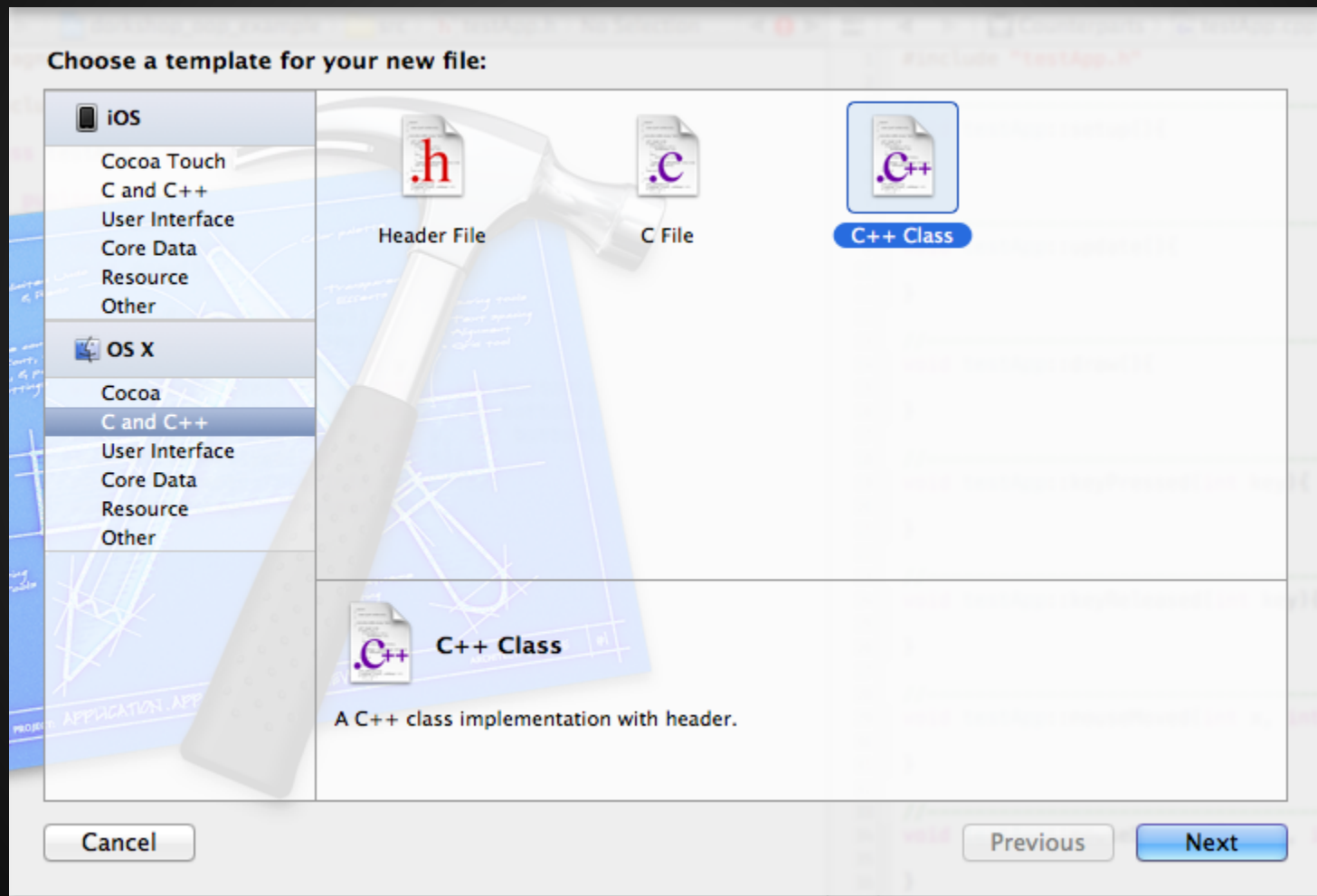
Implementation file: “AwesomeClass.cpp”

- Defines variables and methods

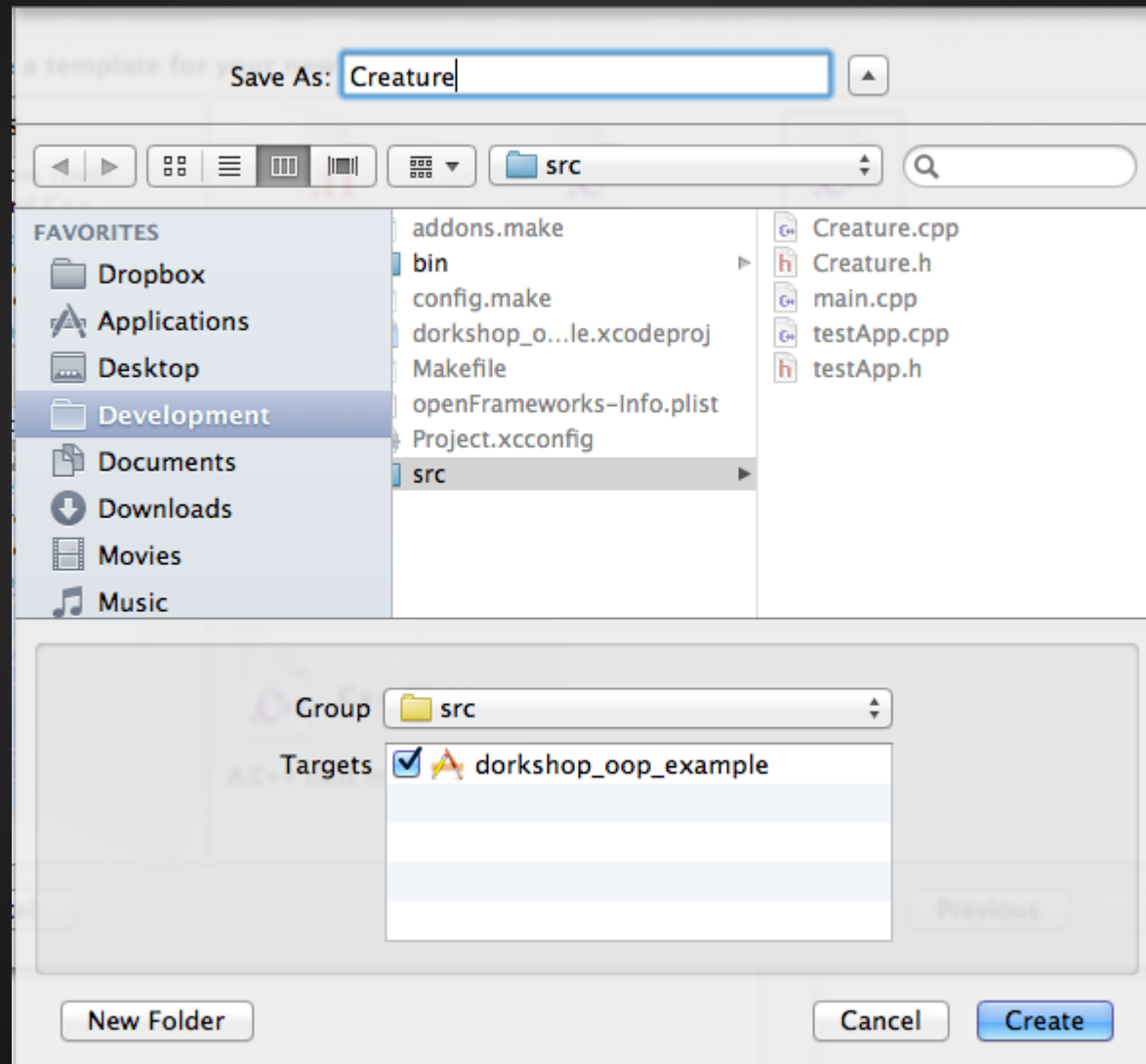
Create a new file in “src”



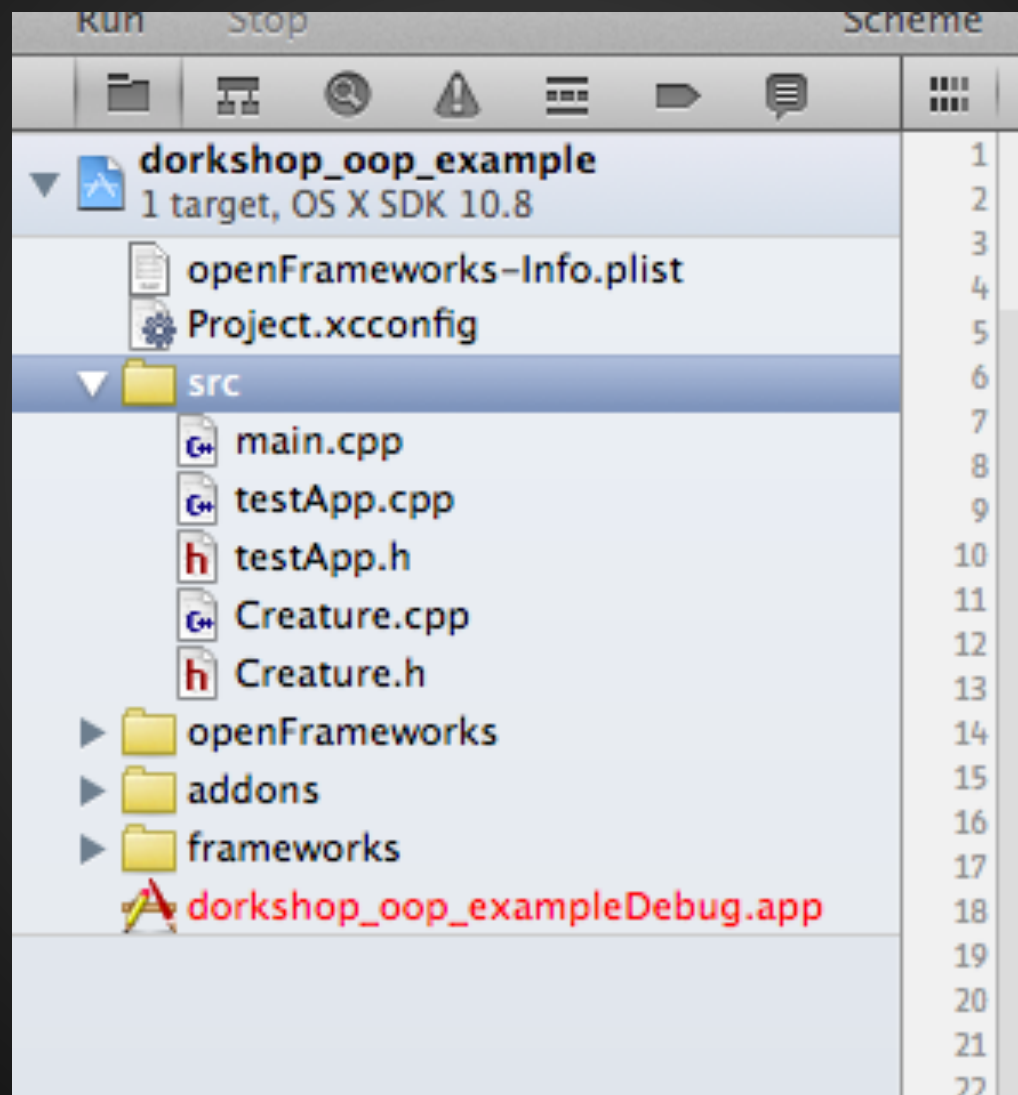
Choose “C++ Class”



Name the file (no extensions)



Like magic, XCode creates both



Link everything up...

```

1  #pragma once
2
3  #include "ofMain.h"
4  #include "Creature.h"
5
6  class testApp : public ofBaseApp{
7
8      public:
9          void setup();
10         void update();
11         void draw();
12
13         void keyPressed(int key);
14         void keyReleased(int key);
15         void mouseMoved(int x, int y);
16         void mouseDragged(int x, int y, int button);
17         void mousePressed(int x, int y, int button);
18         void mouseReleased(int x, int y, int button);
19         void windowResized(int w, int h);
20         void dragEvent(ofDragInfo dragInfo);
21         void gotMessage(ofMessage msg);
22
23 };
24

```

Example code for syntax

http://www.openframeworks.cc/tutorials/first%20steps/003_ooops_object_oriented_programming.html

(just go to www.openframeworks.cc and click on “tutorials” in the navigation bar)