CS 535 Object-Oriented Programming & Design
Fall Semester, 2011
Doc 1 Introduction
Aug 30, 2011

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http://www.opencontent.org/openpub/) license defines the copyright on this document.
References

Wilipedia

Past CS 535 Lecture notes

Reading Assignment

Object-Oriented Design Heuristics, Chapters 1 & 2 for Thursday Sept 2
Course Overview
Course Issues

http://www.eli.sdsu.edu/courses/index.html

Crashing
Course Web Site
Wiki
Screencasts
Prerequisites
Grading
Smalltalk
Goal

Understand how to use classes & objects in code

How to create code that is:

   Understandable
   Modifiable
   Maintainable
   Reusable
Some OO Basics
Why is OO Good?
Does your code achieve those properties of goodness?
Terms

Class
- A blueprint to create objects
- Includes attributes and methods that the created objects all share

Object
- Allocated region of storage
- Both the data and the instructions that operate on that data
class Point
    def initialize(x, y)
        @x = x
        @y = y
    end

    def to_s
        "Point( #{@x},#{@y})"
    end
end

example = Point.new(10,5)
example.to_s

Ruby code. Yes I know most people don't know Ruby. There are lots of variation of syntax and semantics for classes in OO languages.
Abstraction

“Extracting the essential details about an item or group of items, while ignoring the unessential details.”
Edward Berard

“The process of identifying common patterns that have systematic variations; an abstraction represents the common pattern and provides a means for specifying which variation to use.”
Richard Gabriel
Encapsulation

Enclosing all parts of an abstraction within a container
Information Hiding

Hiding of design decisions in a computer program

Hide decisions are most likely to change,
To protect other parts of the program
Class

Represents an abstraction

Encapsulates data and operations of the abstraction

Hide design decisions/details
Heuristics

2.1 All data should be hidden within its class

2.8 A class should capture one and only one key abstraction

2.9 Keep related data and behavior in one place
Non-OO items

Helper methods

Data classes
Helper method

Method in class that
  Does not access any field (data member, instance variables)
  Just uses parameters
class CrosswordPuzzle {

    public void someMethodThatDoesStuff {
        bunch of stuff not shown
        count = vowelCount(aString);
        blah
    }

    private int vowelCount(String word) {
        int vowelCount = 0;
        for (int k = 0; k < word.length(); k++ ) {
            char current = word.charAt(k);
            if ( (current == 'a') || (current == 'e') || (current == 'i') || (current == 'o') || (current == 'u') )
                vowelCount++;
        }
        return vowelCount;
    }
}
**OO Version**

```java
class String {
    public int vowelCount {
        int count = 0;
        for (char current in self)
            if (current.isVowel()) count++;
        return count;
    }
}

class CrosswordPuzzle {
    public void someMethodThatDoesStuff {
        bunch of stuff not shown
        count = aString.vowelCount();
        blah
    }
}
```

Is this better? Why
Data Class

class Point {
    private int x;
    private int y;

    public void setX(int newX) {
        x = newX;
    }

    public int getX() {
        return x;
    }

    public void setY(int newY) {
        y = newY;
    }

    public int getY() {
        return y;
    }
}

Class with get/set methods
constructor
No or very few other methods
Assignment

Pick one of your sizable assignment/projects from another course that uses an OO language.

1. Count the total number of classes and the total number of data classes in the assignment/project.

2. Count the total number of methods and total number of helper methods in the assignment/project.
You will get a lot more out of the class on Thursday if you read the chapters before rather than after the class.