CS 535 Object-Oriented Programming & Design
Fall Semester, 2013
Doc 1 Introduction
Aug 26, 2013

References

Wilipedia

Past CS 535 Lecture notes

Course Overview

Course Issues

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

Crashing
Course Web Site
Wiki
Screencasts
Prerequisites
Grading
Smalltalk

Crash Policy

Will fill all seats in the room

By seniority

- Measured by SDSU credits on transcript
- Alternate undergrad and graduate students
- 2 undegrad students for each graduate student

Provide unofficial transcript

- Hardcopy or softcopy via email
- Available for free in SDSU portal
- Need at least two hours before start of class

Crash Policy

Start adding students in class Thursday

So don't ask for add code an end of class today

No transcript by Thursday assume you have 0 SDSU units

Must be present to get add code

If miss a class then I drop you from crash list

Crash List FAQ

Why not get a bigger room and admit everyone?

No first hard assignment to scare people

No Grader

Do you really want a 500 level class of 120 people?

Crash List FAQ

Sept 9

Last day for regular students to add/drop classes

Open University students have lower priority than SDSU students

For Thursday

Object-Oriented Design Heuristics,
Chapters 1 & 2

Assignment 1

You will get a lot more out of the class on Thursday if you read the chapters before rather than after the class.

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?

For discussion in class

Does your code achieve those properties of goodness?

Why Does Course Still Exist?

Nearly everyone knows & uses object-oriented language

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

From Wikipedia

Example

```
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

Not so much a definition of a class as a goal how we should use a class.

Heuristics

2.1 All data should be hidden within it class

2.8 A class should capture one and only one key abstraction

2.9 Keep related data and behavior in one place

Numbers of the heuristics are from the text Object-Oriented Design Heuristics by Riel

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

Helper Method - Example

```
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

```
class String {

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

Is this better? Why

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

```
class Character {

    public boolean isVowel() {
        return (this == 'a') || (this == 'e' ) || (this == 'i') || (this == "o" )|| (this == "u");
    }
```

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