Object Oriented Programming and Design in Java

Session 1

Instructor: Bert Huang

Today's Plan

- Course information
  - Prerequisites
  - Assignments and expectations
- Course goals

Room Situation

- Tentatively here, 173 Macy, TC
- Register ASAP so CS dept. can request registrar to assign bigger room

Course Information

Staff

- Instructor: Bert Huang
  bert@cs.columbia.edu
  - 3rd year PhD Candidate
    (5th year graduate student)
- TAs: John Graham
  Lauren Pulley
  more TBA

Schedule

Sun	Mon	Tue	Wed	Thu	Fri
John 1-3	Class 11-12:15 Bert 2-4 CEPSR 624	Lauren 5:30-7:30	Class 11-12:15		

COMS W1007

- The second course for majors in computer science. A rigorous treatment of object-oriented concepts using Java as an example language. Development of sound programming and design skills, problem solving and modeling of real world problems from science, engineering, and economics using the object-oriented paradigm.

Topics

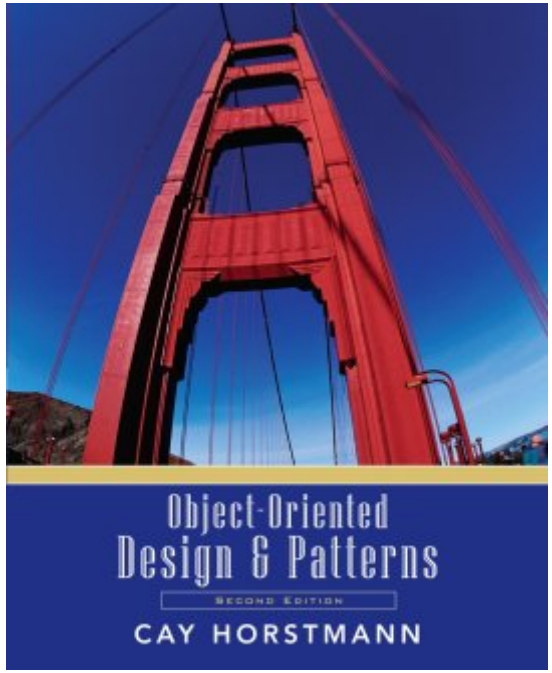
- Programming style
- Classes and methods
- UML diagrams
- Programming by contract
- Designing interfaces
- Polymorphism
- Encapsulation
- Inheritance
- Design patterns
- Frameworks
- Java graphics and GUIs
- Multithreading

Prerequisites

- COMS W1004: Intro to CS and Programming in Java
- or AP CS with a grade of 4 or 5

Required Textbook

- Object Oriented Programming and Design. Cay Horstmann
  - ISBN: 0-471-74487-5



Online Resources

- Course website:
  http://www.cs.columbia.edu/~bert/courses/1007
- Book's website:
  http://www.horstmann.com/design_and_patterns.html
- Courseworks:
  http://courseworks.columbia.edu

Grading

- Five homework assignments, 10% each
  - Late policy: 10% penalty each day (up to three days)
- Midterm exam, 20%
- Final exam, 30%

Programming Assignments

- Programs must compile and run in Java SE 6 (build 1.6.0_17-b04)
  - version currently on CUNIX machines
- We will use graphics, so you'll need an X11-capable machine to run off CUNIX
- You may want to use IDE like *Eclipse*

Academic Honesty

- http://www.cs.columbia.edu/education/honesty
  - 1st offense: grade of zero, report to dean
  - 2nd offense or major violation: fail course, report to dean
- Catching academic dishonesty in CS is easy
- All work in this class is individual; no collaboration

Expectations

- Start early
- Ask questions in class (raise your hand)
  - or office hours, message board, email
- Clarity
- Seek extra help

Course survey

- 1 point extra credit on final grade
- Help me calibrate the course
- http://spreadsheets.google.com/viewform?hl=en&formkey=dGtMam5GMEctQVlsQmpMaFdJUXo2MVE6MA
- (or link off the course homepage)

Java Skills You Should Have

- Compiling a Java program
- Input/output via console or file system
- Basic graphical interface
- commenting for javadoc
- (Programming in an IDE)

Course Goals

Why Study OOP and Design?

- Writing a code is easy
- Understanding code is hard
- Good organization and design of programs makes understanding easier

Bad Design: Goto

```
1. if (stillRunning())
2.   doSomething()
3.   goto 1
```

```
while(stillRunning())
   doSomething()
```

What is OOP?

- A style of programming where all components of computer programs are considered *objects*
  - objects have state, behavior, identity
- **Guidelines and practices to maintain this abstraction intelligently**

Designing a Car Racing Game

- Daunting at first, but identify the objects
  - **Car** objects and a **Track** object interact with a **Physics** object
- then decide responsibilities of classes
  - **Car** stores velocity, size, location. Controls
  - **Track** stores boundaries, start, finish
  - **Physics** determines collisions, adjusts momentum

The Wisdom of OOP

- Breaking down the program into pieces provides bite-sized goals
- By generalizing programs to object interactions, common *patterns* appear
  - Want different types of tracks? **Inheritance** and **polymorphism**

Reading

- Course website/syllabus
  - Survey
- Academic honesty policy
- Horstmann Ch. 1 - Java review