CSC207H: Software Design
Lecture 13

Wael Aboelsaadat

wael@cs.toronto.edu

http://ccnet.utoronto.ca/20075/csc207h1y/

Office: BA 4261

Office hours: R 5-7

Acknowledgement: These slides are based on material by Prof. Karen Reid

What have we studied?

- Tools
- UNIX
- Software QA
- Languages
- Software Design

UNIX

cat cd clear cp

date diff echo head

ls man mkdir more

mv su passwd pwd

rm rmdir sort tail

uniq wc which < | >

Junit and testing

- When to use?
  - static void assertTrue(boolean test)
  - static void assertFalse(boolean test)
  - assertEquals(expected, actual)
  - assertSame(Object expected, Object actual)
  - assertNotSame(Object expected, Object actual)
  - assertNull(Object object)
  - assertNotNull(Object object)
  - fail()

Python

- A small scripting language makes small programs fast to write
- Basic Python:
  - variables, control flow, files
  - sequences: strings, slicing
  - dictionaries, functions and modules
  - Classes & Objects
  - Operators
- Regular expressions

XML

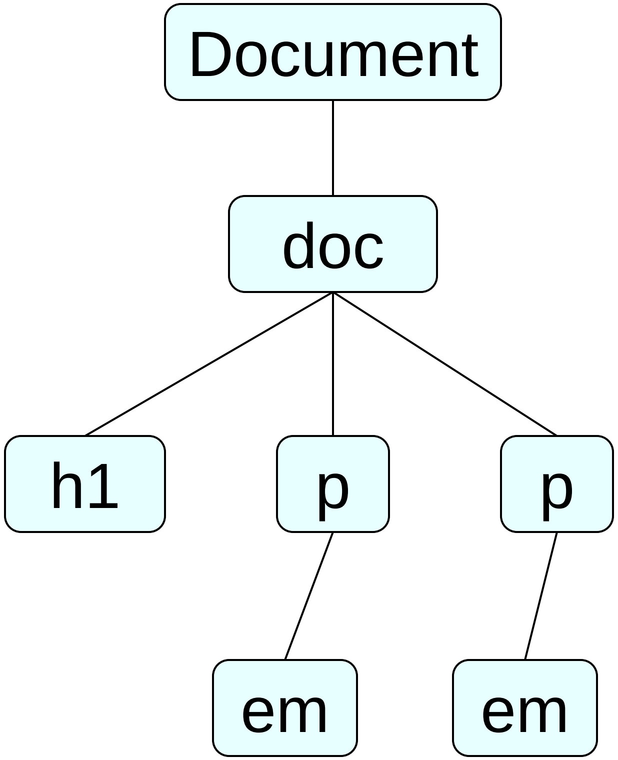
- Valid vs. invalid XML

- DOM way
  - A cross-language API for representing XML documents as trees

XML

- Valid vs. invalid XML

- DOM way



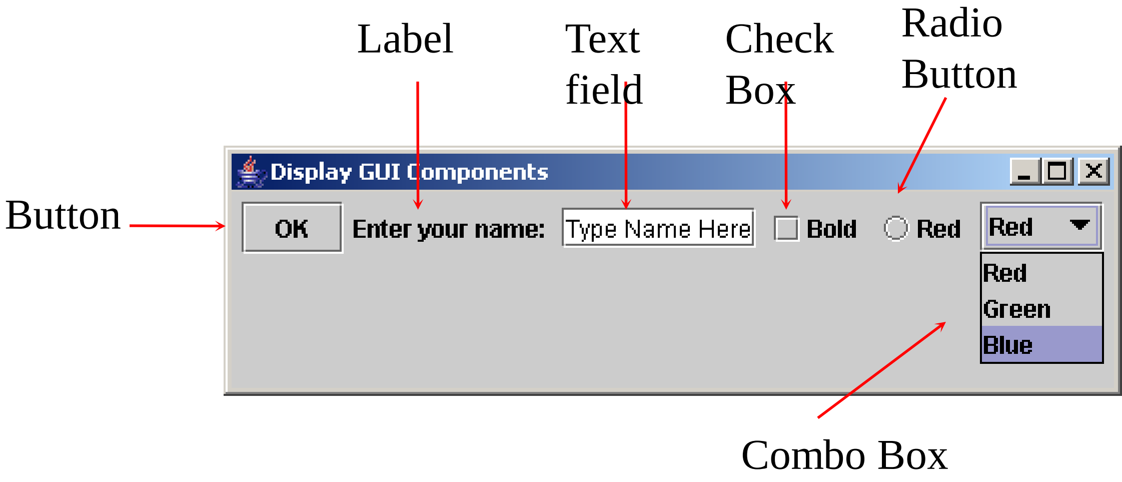
- Input
- <?xml version="1.0" ?>
- <doc>
- <h1>First heading</h1>
- <p>First
- <em>paragraph</em>.</p>
- <p><em>Second
- paragraph.</em></p>
- </doc>

- Output h1 p p

Java

- New features
  - maps
  - Auto boxing
  - generics
- JDom
- Reflection

- GUI/event-driven programming



Label

Text field

Check Box

Radio Button

Button

Combo Box

Class Design

- How to identify classes?

- What makes a good design?

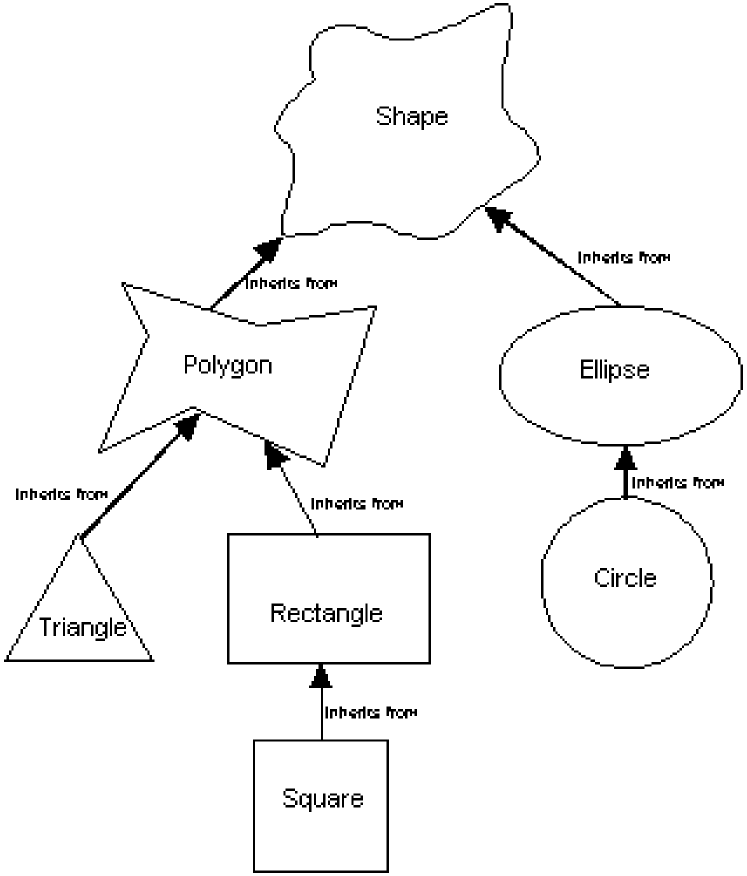
- What are the possible relations between classes?

Design Example 1

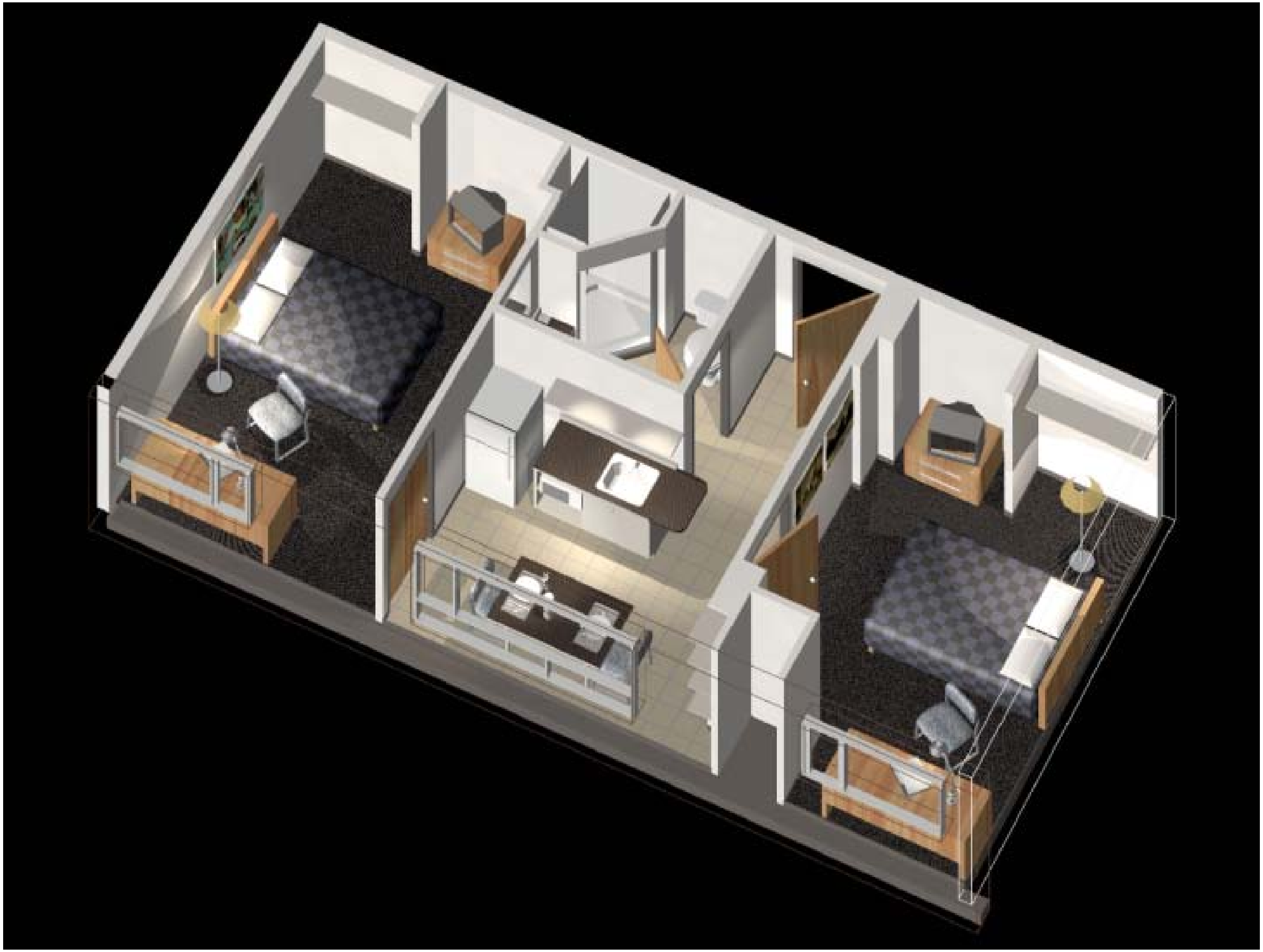
- Circle, rectangle, triangle, polygon, square, ellipse

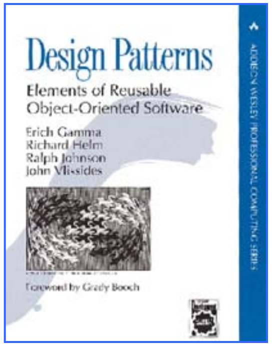
Design Example 1

- Circle, rectangle, triangle, polygon, square, ellipse



Design Example 2





Design Patterns

Elements of Reusable Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Foreword by Grady Booch

ADDISON WESLEY PROFESSIONAL COMPUTING SERIES

Design Pattern

Creational	**Structural**	**Behavioral**
Factory Method	Adapter	Interpreter
Abstract Factory	Bridge	→ Template Method
Builder	→ Composite	Chain of Responsibility
Prototype	Decorator	→ Command
→ Singleton	Flyweight	→ Iterator
	Facade	Mediator
	Proxy	Memento
		→ Observer
		State
		Strategy
		→ Visitor

Design Example 2

- Singleton
- Composite
- Template Method
- Command
- Iterator
- Observer
- Visitor

What is on the exam?

- Everything on the exam we have talked about in class, was covered on an assignment, or is in the lecture notes

- We will not ask you to take the integral of a function!

- We will not expect you to know any obscure API

- Distribution:
  - Java/python 40%
  - Software design 40%
  - Misc. 20%

The exam aid sheet

- 1 Aid sheet double sided standard-size (A4)
  - You can write anything on that paper!
  - No electronic devices of any sort
    - Calculators, cell phones, pdas, laptops, mp3 players, pagers, cameras... *not even if turned off*

What should you include on the aid sheet?

- Organization is key

- Include
  - notes
  - Assignment code snippets
  - Some example code
  - Some pieces of the Java/Python API would be helpful