Final Words

Hsuan-Tien Lin

Dept. of CSIE, NTU

June 13, 2013

Before the Midterm

- Basic Java/OOP
- Classes/Objects
- Arrays
- Inheritance
- Polymorphism

After the Midterm

- Exception
- Interface
- Swing/Inner Class
- Thread
- Generics
- I/O
- Design Patterns

Design Patterns

A Pattern is a solution to a problem in a context.

- Strategy, Template Method, State
- Decorator, Adapter, Facade, Proxy
- Factory Method, Abstract Factory, Singleton
- Observer, Command
- Iterator, Composite
- Model-View-Controller

S.O.L.I.D. Principles

- Single Responsibility: “a class should have only a single responsibility” (abstraction)
- Open/Closed: “software entities should be open for extension, but closed for modification” (polymorphism)
- Liskov Substitution: “objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program” (inheritance)
- Interface Segregation: “no client should be forced to depend on methods it does not use” (i.e. program to interfaces)
- Dependency Inversion: “abstractions should not depend upon details; details should depend upon abstractions.” (implicit in many design patterns)

Three To-Be-Improved Things in the Course (1/3)

加強： Interaction of Class

- **interaction level went down** as class progressed
- more interaction can lead to better learning!

Three To-Be-Improved Things in the Course (2/3)

加強： Quality of Homework

- homework often designed in **super hurry** and hence needs patches and clarifications
- my apology for all the inconveniences!

Three To-Be-Improved Things in the Course (2/3)

加強： Performance of Teaching

- instructor is often **too exhausted to teach perfectly**, partially because of the huge teaching loads in the same semester
- my apology again for not doing better!

Four Excellent Things in the Course (1/4)

很好: Hard Working Students

- challenging homework/class, but **many of you work hard**
- very good homework performance, beyond expectation!

Four Excellent Things in the Course (2/4)

很好: Broad Contents

- wanted to make this course **OOP** instead of only Java
- covered much more than the old days, with many angles of OOP

Four Excellent Things in the Course (3/4)

很好: Super TAs

- many homework programs with the super-large class—**very difficult to grade**
- TAs: enthusiastic and worked lots more than needed
  —e.g. the GIT idea from the TAs worked quite nicely

Four Excellent Things in the Course (4/4)

很好： Active Instructor

- high pressure to try to teach the super-large class
  —**especially when students don't laugh when hearing jokes**
- instructor still tries to be active and very persistent in telling jokes

How to Be an Excellent Programmer?

ability / characteristic	related courses
sincere, communication, work hard, ...	
domain knowledge	
language feature • workable program • bug-free program e.g., C# property, DateTime.Now • most suitable feature e.g., if vs. switch statements	Programming Language (Compiler) (Operating System) (Computer Architecture)
program behavior	Data Structure Algorithm
API, library • simple API e.g., Date, Calendar, GregorianCalendar • complex API, with design concept e.g., swing's event/listener, RMI	
experience • write program by yourself • trace other's real & good program	(Data Structure) (Algorithm)

Most Important of All: Never Stop Learning!

http://www.books.com.tw/exep/prod/booksfile.php?item=0010464414

THANK YOU!!