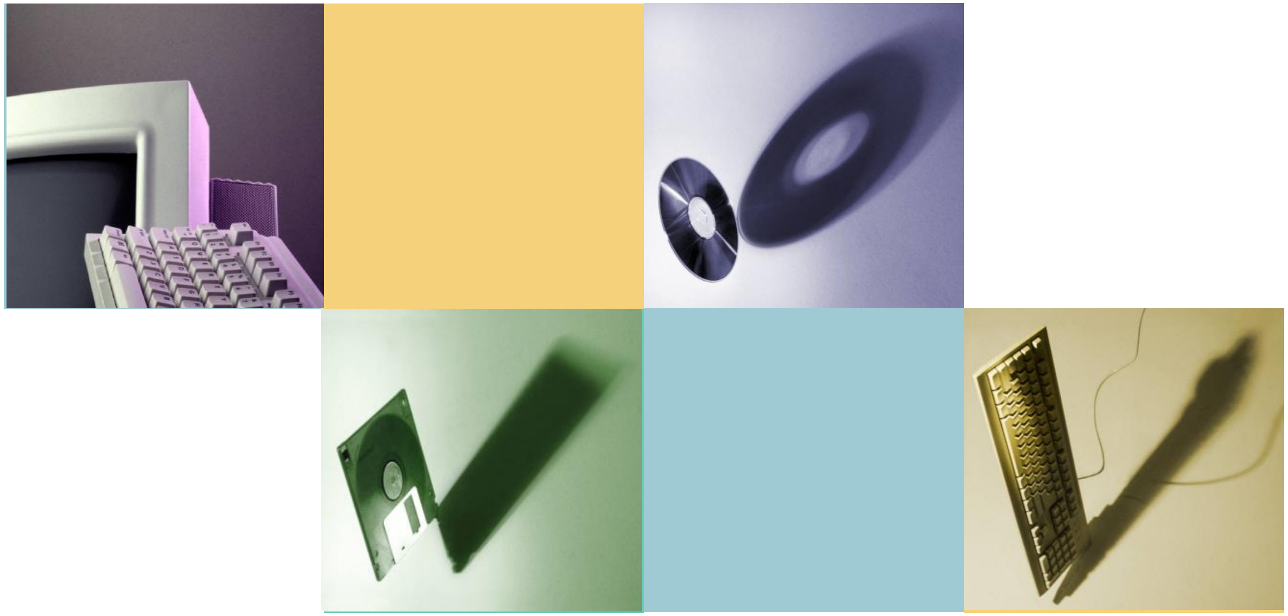
Object-Oriented Programming



Chuan-Kang Ting

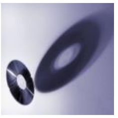
Dept of Computer Science and Information Engineering
National Chung Cheng University

About the Course (1)

- **Time and place**
  - Tue/Thu **13:15**~14:30, EA101
- **Office hours**
  - Tue 14:30~16:30
- **Evaluation**

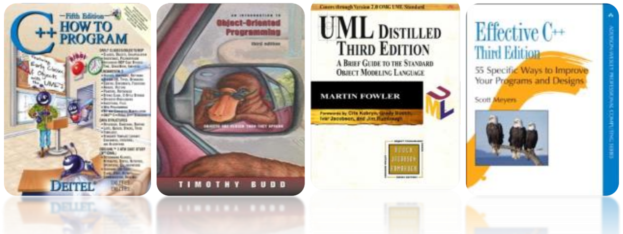
– Assignments:	30%
– Midterm Exam:	30%
– Final Exam:	40%

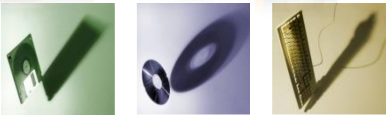
All materials and related information will be announced on **eCourse**



About the Course (2)

- **Textbook**
  - Walter Savitch, "**Absolute C++**", Addison Wesley (開發)
- **References**
  - H. Deitel and P. Deitel, "C++ How to Program", Prentice Hill, 2005
  - T. Budd, "An Introduction to Object-Oriented Programming", Addison Wesley, 2002
  - S. Meyers, "Effective C++: 55 Specific Ways to Improve Your Programs and Designs", Addison Wesley, 2005
  - M. Fowler and K. Scott, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Addison Wesley, 2003


C++ HOW TO PROGRAM
Fifth Edition
DEITEL
An Introduction to Object-Oriented Programming
third edition
TIMOTHY BUDD
UML DISTILLED
THIRD EDITION
A BRIEF GUIDE TO THE STANDARD OBJECT MODELING LANGUAGE
MARTIN FOWLER
Effective C++
Third Edition
55 Specific Ways to Improve Your Programs and Designs
Scott Meyers



About the Course (3)

- **Instructor**
  - Dr. rer. nat. **Chuan-Kang Ting (丁川康)**
    - Dr. rer. nat. (Doctor of Nature Science)
    - from University of Paderborn, Germany (2002 Oct – 2005 Oct)
  - Email: ckting@cs.ccu.edu.tw
  - Office: EA 506
- **TA**
  - 王廷禎、溫育瑋、沈庭聿
  - Office: EA 505

Academic Honesty

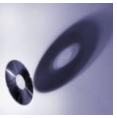
All work done on assignments, midterms and finals should be your own work. Cheating on any kind of assignment and examination will be taken very seriously.

Any such incident will result in a letter describing the incident which is placed in your file on campus.
Penalties will also be imposed by the department and the university.

Very severe incidents of academic dishonesty can result in suspension or expulsion from the university.

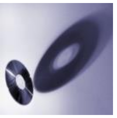
Important Notes

- **100% Academic honesty**
  - I mean it!
  - All assignments will be carefully checked for plagiarism
- **Some fundamentals of C won’t be repeated**
  - You MUST know them; otherwise, enhance it
- **Additional/make-up lessons**
  - 1 or 2 nights
- **Team programming**
  - Including peer scoring
- **No photo/video** 🚫



Chapter 0

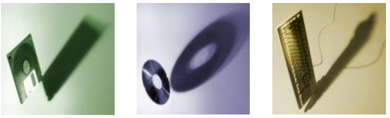
Introduction



Programming

You have learned…

You will learn from this course



Background Ideas (1)

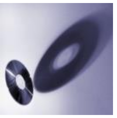
- **What are “objects”?**
  - Objects in real world:
    people, animals, plants, cars, buildings, computers, …
    - They all have attributes (e.g., size, shape, and color)
    - They all exhibit behaviors (e.g., a car accelerates and brakes)
  - Objects in software:
    Are essentially reusable software components that model items in the real world

Background Ideas (2)

- **Why "object-oriented"?**
  - When you are required to modify/upgrade the display part of a PC…
    - All-in-one main board
    - Modern component-wise structure
  - Similar scenario in modifying/rewriting some functionality of a program
    - Structural program
    - Object-oriented program → Soft IC

OOP

- **The characteristics of OOP – Pie**
  - Encapsulation
    - information hiding and abstraction
  - Inheritance
    - code reusability
  - Polymorphism
    - a single name with multiple meanings in inheritance

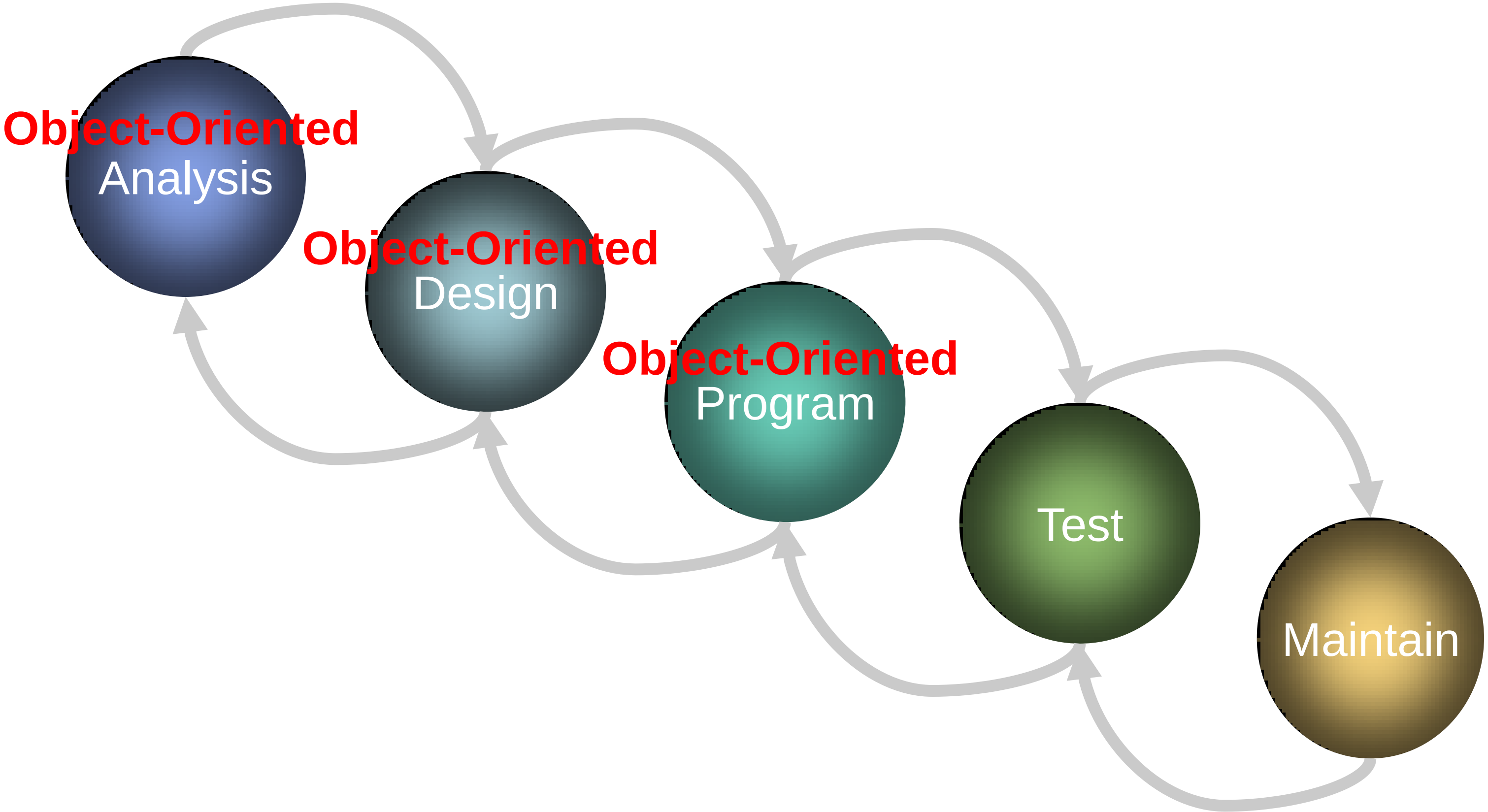


OO Phases

- **OOA**
  - Object-Oriented Analysis
- **OOD**
  - Object-Oriented Design
- **OOP**
  - Object-Oriented Programming

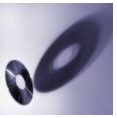
Software Engineering

- **Iterative model** (cf. waterfall model)



Topics

1. C++ Basics
2. Flow of Control
3. Function Basics
4. Parameters and Overloading
5. Arrays
6. **Structures and Classes**
7. Constructors
8. Operator Overloading, Friends, and References
9. Strings
10. Pointers and Dynamic Arrays
11. Separate Compilation and Namespaces
12. Streams and File I/O
13. Recursion
14. **Inheritance**
15. **Polymorphism and Virtual Functions**
16. Templates
17. Linked Data Structures
18. Exception Handling
19. Standard Template Library
20. **Patterns and UML**



The Course

Centers on “OO” and the stuff “outside” braces

- Fundamentals and philosophy
- Programming skills
- Software engineering

To be Successful

- Attend the lectures
- Develop skills –

  Write a LOT of programs
- Read the chapters
- Think and ask questions
- Make use of TAs

(from Daugherity, Petersen, and Stroustrup's "Computer Programming using C++")