# 計算機概論
# Computer Science

**課程編號：901 10110 -04**
**科目名稱：計算機概論**
**授課教師：黃鼎偉**
**時間地點：二5三78 電機明達館231**

## Course Information (I)

- Class homepage
  - http://iphotonics.dyndns.org/dwhuang/courses/cs/
- Time/Location: Tuesday 5,Wednesday 78, MD 231
- Instructor: Ding-wei Huang 黃鼎偉 (dwhuang@cc.ee.ntu.edu.tw) Office: MD 720
- TA:
  - Office hour:
- Credits: 3 units
- Textbooks: J. Glenn Brookshear, Computer Science, An Overview 10th Ed, Pearson Addison-Wesley 2009.



## Course Information (II)

- Grading
  - Midterm exam: 35%
  - Final exam: 40%
  - Homework: 25%
  - Project Bonus: 1-5%
- Some other notes
  - Homework and its due date will be announced on the class homepage.
  - Please hand-in your homework on time. For the delay of your homework, 20 points/day will be subtracted till zero.

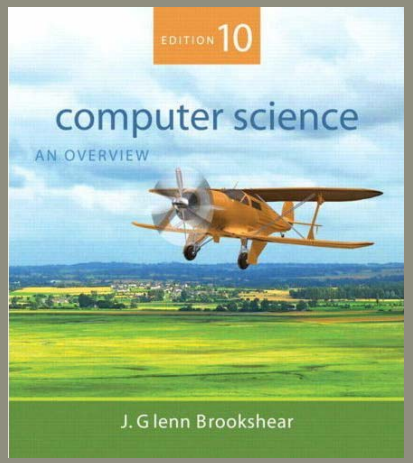


**國立臺灣大學九十七學年度　第二學期教學計劃表**

| 一、科目名稱： 計算機概論 課程編號 901 10110 -04 | | | | |
|---|---|---|---|---|
| 二、教學內容與進度： | | | | |
| 週次 | 上課日期 | 教材單元與進度 | 評量與檢討 | 備註 |
| 一 | 02/23,24 | Introduction | | 加退選 |
| 二 | 03/02,03 | Topic 1 | | 加退選 |
| 三 | 03/09,10 | Topic 1/2 | | 加退選 |
| 四 | 03/16,17 | Topic 2/3 | | |
| 五 | 03/23,24 | Topic 3 | | |
| 六 | 03/30,31 | Topic 3/4 | | |
| 七 | 04/06,07 | 停課一次 | 溫書假 | |
| 八 | 04/13,14 | Topic 4 | | |
| 九 | 04/20,04/21 | 04/21 Midterm Exam | 期中考 35% | |
| 十 | 04/27,28 | Topic 5 | | |
| 十一 | 05/04,05 | Topic 6 | | |
| 十二 | 05/11,12 | Topic 7 | | |
| 十三 | 05/18,19 | Topic 8 | | |
| 十四 | 05/25,26 | Topic 9 | | |
| 十五 | 06/01,02 | Topic 10 | | |
| 十六 | 06/08,09 | Topic 11 | | |
| 十七 | 06/15,16 | Topic 12 | | |
| 十八 | 06/22,06/23 | 06/23 Final Exam | 期末考 40% | 作業25% |

# Chapter 0

# Introduction




## Chapter 0: Introduction

- 0.1 The Role of Algorithms
- 0.2 The Origins of Computing Machines
- 0.3 The Science of Algorithms
- 0.4 Abstraction
- 0.5 An Outline of Our Study
- 0.6 Social Repercussions



## Terminology

- **Algorithm:** A set of steps that defines how a task is performed
- **Program:** A representation of an algorithm
- **Programming:** The process of developing a program
- **Software:** Programs and algorithms.
- **Hardware:** Equipment



## Figure 0.1 An algorithm for a magic trick

**Effect:** The performer places some cards from a normal deck of playing cards face down on a table and mixes them thoroughly while spreading them out on the table. Then, as the audience requests either red or black cards, the performer turns over cards of the requested color.

**Secret and Patter:**

Step 1. From a normal deck of cards, select ten red cards and ten black cards. Deal these cards face up in two piles on the table according to color.

Step 2. Announce that you have selected some red cards and some black cards.

Step 3. Pick up the red cards. Under the pretense of aligning them into a small deck, hold them face down in your left hand and, with the thumb and first finger of your right hand, pull back on each end of the deck so that each card is given a slightly *backward* curve. Then place the deck of red cards face down on the table as you say, "Here are the red cards in this stack."

Step 4. Pick up the black cards. In a manner similar to that in step 3, give these cards a slight *forward* curve. Then return these cards to the table in a face-down deck as you say, "And here are the black cards in this stack."

Step 5. Immediately after returning the black cards to the table, use both hands to mix the red and black cards (still face down) as you spread them out on the tabletop. Explain that you are thoroughly mixing the cards.

Step 6. As long as there are face-down cards on the table, repeatedly execute the following steps:

6.1. Ask the audience to request either a red or a black card.

6.2. If the color requested is red and there is a face-down card with a concave appearance, turn over such a card while saying, "Here is a red card."

6.3. If the color requested is black and there is a face-down card with a convex appearance, turn over such a card while saying, "Here is a black card."

6.4. Otherwise, state that there are no more cards of the requested color and turn over the remaining cards to prove your claim.

## History of Algorithms

- The study of algorithms was originally a subject in mathematics.
- Early examples of algorithms
  - Long division algorithm
  - Euclidean Algorithm
- **Gödel's Incompleteness Theorem**: Some problems cannot be solved by algorithms.



## Figure 0.2 The Euclidean algorithm

**Description:** This algorithm assumes that its input consists of two positive integers and proceeds to compute the greatest common divisor of these two values.

**Procedure:**

Step 1. Assign M and N the value of the larger and smaller of the two input values, respectively.

Step 2. Divide M by N, and call the remainder R.

Step 3. If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N.



## Origins of Computing Machines

- Early computing devices
  - Abacus: positions of beads represent numbers
  - Gear-based machines (1600s-1800s)
    - Positions of gears represent numbers
    - Blaise Pascal, Wilhelm Leibniz, Charles Babbage



## Figure 0.3 An Abacus

## Early Data Storage

- Punched cards
  - First used in Jacquard Loom (1801) to store patterns for weaving cloth
  - Stored programs in Babbage's Analytical Engine
  - Popular through the 1970's
- Gear positions



## Early Computers

- Based on mechanical relays
  - 1940: Stibitz at Bell Laboratories
  - 1944: Mark I: Howard Aiken and IBM at Harvard
- Based on vacuum tubes
  - 1937-1941: Atanasoff-Berry at Iowa State
  - 1940s: Colossus: secret German code-breaker
  - 1940s: ENIAC: Mauchly & Eckert at U. of Penn.



## **Figure 0.4** The Mark I computer



## Personal Computers

- First used by hobbyists
- IBM introduced the PC in 1981
  - Accepted by business
  - Became the standard hardware design for most desktop computers
  - Most PCs use software from Microsoft

## Computer Science

- The science of algorithms
- Draws from other subjects, including
  - Mathematics
  - Engineering
  - Psychology
  - Business Administration
  - Psychology

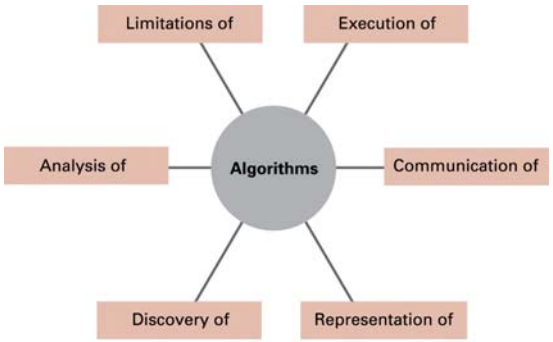


## Central Questions of Computer Science

- Which problems can be solved by algorithmic processes?
- How can algorithm discovery be made easier?
- How can techniques of representing and communicating algorithms be improved?
- How can our knowledge of algorithms and technology be applied to provide better machines?
- How can characteristics of different algorithms be analyzed and compared?



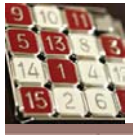
## Figure 0.5 The central role of algorithms in computer science






## Abstraction

- **Abstraction:** The distinction between the external properties of an entity and the details of the entity's internal composition
- **Abstract tool:** A "component" that can be used without concern for the component's internal properties



## Outline of Our Study

- Chapter 1: Data Storage
- Chapter 2: Data Manipulation
- Chapter 3: Operating Systems
- Chapter 4: Networks and the Internet
- Chapter 5: Algorithms
- Chapter 6: Programming Languages



## Outline of Our Study (continued)

- Chapter 7: Software Engineering
- Chapter 8: Data Abstractions
- Chapter 9: Database Systems
- Chapter 10: Computer Graphics
- Chapter 11: Artificial Intelligence
- Chapter 12: Theory of Computation



## Social Repercussions

- Advances in computer science raise new questions.
  - In law: Questions of rights and liabilities
  - In government: Questions of regulation
  - In the work place: Questions of professionalism
  - In society: Questions of social behavior



## Ethical Theories

- Consequence based:
  What leads to the greatest benefit?
- Duty based:
  What are my intrinsic obligations?
- Contract based:
  What contracts must I honor?
- Character based:
  Who do I want to be?