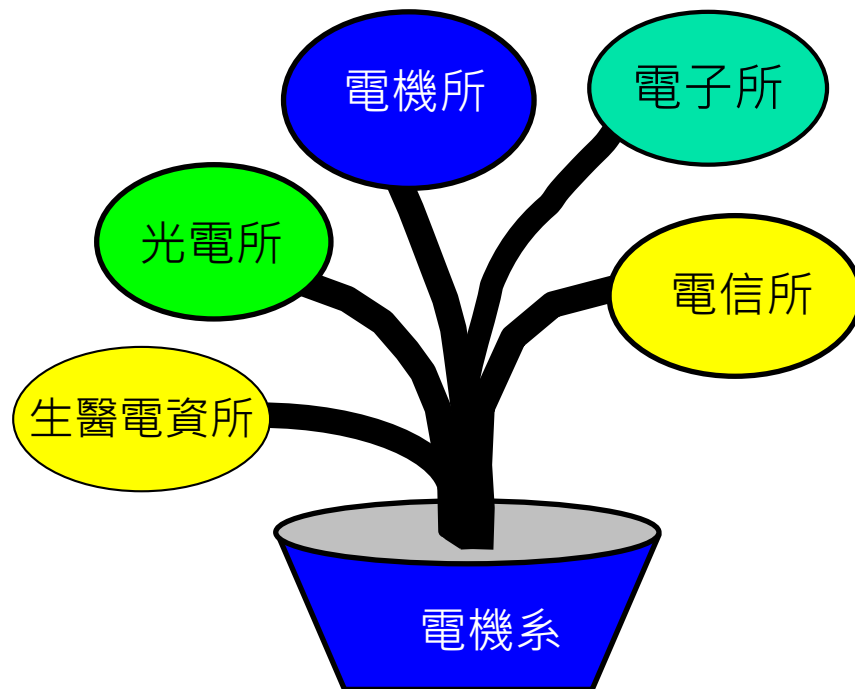# Computer Programming

Course Information

Hung-Yun Hsieh
September 6, 2022

What is This Course About?

- Computer Programming
  - Introduction to computer programming for students who may have little or no prior programming experience
  - Students are expected to design, write, and debug *sophisticated* computer programs by the end of this course

  - As an introductory course, this course is the foundation for continuing curriculum on
    - data structure (資料結構)
    - algorithm (演算法)
    - systems programming (系統程式設計)
    - operating systems (作業系統)
    - …

Is Programming Important to EE?

- Yes, even if you will <u>not</u> major in CS

- 電機所
  自動控制、電力、計算機

- 光電所
  光電

- 電信所
  電波、通訊與信號處理、
  資料科學與智慧網路

- 電子所
  積體電路設計、奈米電子、
  電子設計自動化

- 生醫電資所
  醫學工程

- ☞ All require programming
  - Programming allows us to
    command computers _as we wish_

Which Language?

- C/C++
  - ☞ C++ is evolved from C and both are popularly used today
  - C/C++ is a high-level programming language, but has the merits of a low-level language (efficiency)
  - C/C++ is a general-purpose programming language, but is used widely for scientific and engineering applications
- Many of us will learn more than one language
  - ☞ Many modern languages are created with C/C++ in mind
  - ☞ C/C++ can act as a stepping stone for learning other languages such as Java (Android) and Objective-C (iOS)
  - ☞ C/C++ can work in companion with scripting languages such as Python very well

What Will be Covered?

- Course plan
  - Introduction
  - Variable and operation
  - Control structure
  - Array and pointer
  - Function and library

  Function-oriented programming (C/C++)

  - Class
  - Operator overloading
  - Inheritance
  - Polymorphism
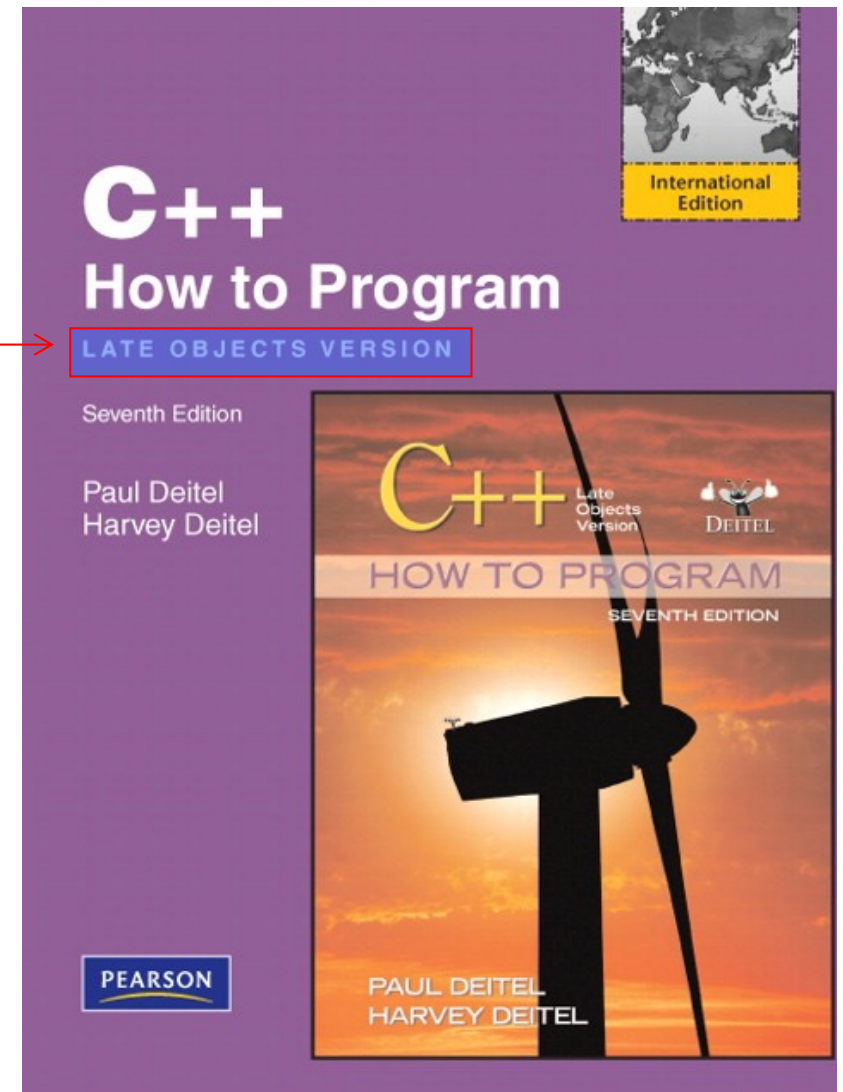 - Miscellaneous topics

  Object-oriented programming (C++)

Is There a Textbook?

- **Yes and no**
  - ☞ Deitel & Deitel, "C++ How to Program: Late Objects Version," 7th edition, Prentice Hall, 2010.

- **But note…**
  - We *will not* follow this book section-by-section
  - Slide handouts will be provided before class
  - Textbook *is not required*, but it can supplement the slides *with details*

How about Grading?

- **Grading**
  - First midterm exam (25%)
  - Second midterm exam (25%) ⎫ 上機考
  - Homework assignments (20%)
    - Weekly assignments (preview)
    - Supplementary assignments
  - Term project (30%)
    - A team-based hands-on project (demonstration required)

1 ↓ ↓ ↓ 80+?

| A+ | A | A- | B+ ~ F |

☞ 等第制分數非原始分數直接對應

Weekly Assignments

☞ Weekly assignments mostly include preview questions

■ Why doing previews?

  ■ Sometimes it is difficult to immediately grasp new materials during the class

  ■ Yet course materials build upon each other

  ■ Previews allow us to use class hours more *efficiently*

☞ How to do previews?

  ■ 於課前預習上課投影片，了解上課範圍

  ■ 於課前「成功執行」投影片程式 (解決編譯問題)

  ■ 對投影片內容不清楚之處，參照書本或與同學討論

  ■ 完成課前預習作業

Slides with Examples

■ **Conditional operator (ternary operator)**

```
expr1 ? expr2 : expr3
```

> If expr1 is true, expr2 is evaluated;
> if expr1 is false, expr3 is evaluated

☞ The return value of the ?: expression is equal to the
value of the expression (**expr2** or **expr3**) evaluated

```cpp
#include <iostream>
using namespace std;

int main( )
{
    double x=0,y=0;
    cout<<"Enter a number: ";
    cin>>x;

    y = (x>0) ? (x) : (-1*x);
    cout<<"Absolute value is "<<y<<endl;
}
```

> How to rewrite the program
> using *if* and *else*?

> There is a **sequence point**
> after the evaluation of the
> condition (expr1)

語法

Try it

範例程式

Find out what this is

Weekly Programming Sessions

- **Programming session (上機)**
  - Programming sessions are an integral part of this course
  - Practice makes perfect!

- **Computer Programming Laboratory (CPL)**
  - You will participate in programming labs that can help you master the topics covered in this course
  - It is to your advantage to take the CPL course along with this course
  - Tuesday afternoon from 14:20 to 16:20

More on CPL

- **Computer Programming Laboratory**
  - You get extra 2 credits for taking this course
  - This course cannot be taken without taking the *Computer Programming* course by the same professor

- **Grading**
  - First midterm exam (25%)
  - Second midterm exam (25%) } 上機考
  - Weekly programming session (50%)
    - Programming labs
    - Term project tools

A Note on the Course Platform

- **NTU COOL**
  - https://cool.ntu.edu.tw/courses/16182
  - 重要公告
  - 預習作業
  - 預習投影片討論
  - 課後投影片

- **Online Judge**
 - https://mozart.ee.ntu.edu.tw
  - 上機 & 考試

Supplementary Assignments

☞ Programming is not just about *syntax*

- Weekly assignments allow you to get hold of the syntax
- Syntax is necessary, but not sufficient
- How a task can be completed step by step (*algorithm*) is also important (and *more difficult* to master)

■ Supplementary assignments

- About 1~2 assignments in one semester
- It involves more comprehensive use of the programming techniques learned
- It involves the use of algorithms for solving the problem
- Helpful for preparing for the term project

Sample Homework Problems

(20%)[Mouse maze] [Required File: HW2_p5.cpp]

Write a program that simulates a mouse in a maze as follows. The entrance spot, where the mouse starts its journey, is <u>chosen by the user</u> who runs the program. The maze is given as follows:
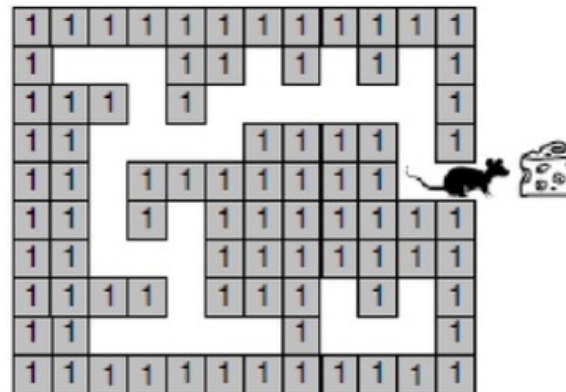
Each element of the array can be black or white. A black element means the wall (the mouse cannot pass), and a white element can be used by the mouse.

The value of the integral can be obtained if there is an interval for integration (i.e. [a, b]). In this problem, write some functions to help us find the **integral** of a polynomial equation, and the **value of integral** on an interval.

(X,Y)

Your program should let user input the y coordinates of pig's location) then or equal to 0) and angle θ (degree) coordinates is the initial point of the uadrant, so you must output an error rdinate.

Will I Master C/C++ After All These?

- Not quite…
  - You can master C/C++ only after you have used it to solve a real problem that you *did not believe you can*
  - Term project is the most <u>defining</u> part

- Term project
  - Teamwork (2~3 students per team)
  - Key factors: innovation (proposal & framework), execution (coding & debugging), and package (demo & presentation)
  - You will be guided towards the project as the course goes on

More on Term Project

- **Term project**
  - It is scheduled to kick off *after the second midterm exam*
  - You will have learned everything about functional-oriented programming (C++ without classes) by then

  - Term project needs to be implemented with object-oriented programming (C++)

  - We plan to use *Week 16* for demo so you have 5-6 weeks for doing the term project

Sample Term Projects

Got Questions?

- **聯絡老師**
  - 研究室：電機二館 546 室
  - Email: hungyun@ntu.edu.tw
  - Office Hour: 有任何問題歡迎寄信給老師約時間

- **本班助教**
  - 實驗室：博理館 521 室
  - Email: cplta2022@gmail.com
  - 帶領上機課、批改作業並協助解答課程相關問題

今日上機課 (14:20-16:20)

- 學新館 414 電腦教室
  - 完成今日的課程介紹
    - 期末專題
  - 帶領大家初步了解程式開發環境 (IDE)
    - First C++ Program
  - 介紹線上批改系統 (Online Judge)
    - 上機練習 & 上機考
  - 協助解決個人電腦安裝 IDE 問題

- 自下週起 (9/13)，**星期二第 6 節**改於學新館 414 電腦教室上課
  - 星期三第 89 節仍於電二 143 室上課

19