Course: Introduction to Computer Science

國立中正大學資訊工程系
副教授 陳裕賢 (Yuh-Shyan Chen)

Homepage:  http://www.cs.ccu.edu.tw/~yschen
E-mail:         yschen@cs.ccu.edu.tw
分機:            33115
Office:          EA506

Text book

- C How To Program (Introducing C++
  and Java)
  - DEITEL & DEITEL
  - 全華書局

- My homepage

學期成績計分方式

- 期中考 (30%)
- 期末考 (30%)

- Homeworks (40%)
  - 計算機實習
    - 分四個實習班級
  - 每週有週的小作業作業於計算機實習課作
  - 一學期有兩個較大的程式 (projects)
    - 分別於期中與期末時繳交

Chapter 1

Introduction to Computers and C++ Programming

Associate Prof. Yuh-Shyan Chen
Dept. of Computer Science and Information Engineering
National Chung-Cheng University
1.1 Introduction

- We will learn the C programming language
  - Learn structured programming and proper programming techniques
  - Chapter 15 - Introduction to C++, a superset of C (more later)

- This course is appropriate for
  - Technically oriented people with little or no programming experience
  - Experienced programmers who want a deep and rigorous treatment of the language

This text provides an introduction to programming in the version of C standardized in 1989
- Through ANSI (American National Standards Institute)
- And worldwide through ISO (International Standards Organization)
- In 1999
  - ISO approved a new version of C (C99)

C++ and Java
- Object-oriented programming language based on C
1.2 What is a Computer?

- **Computer**
  - Device capable of performing computations and making logical decisions
  - Computers process data under the control of sets of instructions called computer programs

- **Hardware**
  - Various devices comprising a computer
  - Keyboard, screen, mouse, disks, memory, CD-ROM, and processing units

- **Software**
  - Programs that run on a computer

1.3 Computer Organization

- **Six logical units in every computer:**
  1. **Input unit**
  2. **Output unit**
  3. **Memory unit**
  4. **Arithmetic and logic unit (ALU)**
  5. **Central processing unit (CPU)**
  6. **Secondary storage unit**

- **Computer Organization**

  - Input Unit
  - Output Unit
  - Storage Unit
  - Processing Unit
  - Control Unit

- **Operating Systems (Systems Program)**

  - **Applications**

  - **Hardware**
1.4 Evolution of Operating Systems

- Batch processing
  - Do only one job or task at a time
- Operating systems
  - Manage transitions between jobs
  - Increased throughput - amount of work computers process
- Multiprogramming
  - Many jobs or tasks sharing the computer resources
- Timesharing
  - Runs a small portion of one user’s job then moves on to service the next user

1.5 Personal Computing, Distributed Computing, and Client/Server Computing

- Personal computers
  - Economical enough for individual
- Distributed computing
  - Computing distributed over networks
- Client/server computing
  - Sharing of information across computer networks between file servers and clients (personal computers)

1.6 Machine Languages, Assembly Languages, and High-level Languages

1. Machine languages
   - Strings of numbers giving machine specific instructions
   - Example:
     +130042774
     +1400593419
     +1200274027

2. Assembly languages
   - English-like abbreviations representing elementary computer operations (translated via assemblers)
   - Example:
     LOAD BASEPAY
     ADD OVERPAY
     STORE GROSSPAY

3. High-level languages
   - Similar to everyday English and use mathematical notations (translated via compilers)
   - Example:
     grossPay = basePay + overtimePay
A Program in C

```c
enum fruit_tea {apple, banana, orange};
fruit_tea taste;
if (taste == apple)
    printf("Taste is Apple \n");
else if (taste == banana)
    printf("Taste is Banana \n");
else
    printf("Taste is Orange");
```

A Program

- A set of instructions

Example:

```c
enum fruit_tea {apple, banana, orange};
fruit_tea taste;
if (taste == apple)
    printf("Taste is Apple \n");
else if (taste == banana)
    printf("Taste is Banana \n");
else
    printf("Taste is Orange");
```

Language

- Syntax
  - Grammars, 句型, 架構
- Semantics
  - 語意, 定義

Example:

I am a boy.
I am a girl.

Program Execution

- CPU (Control Unit and ALU)
- Monitor
- Hard disk (Secondary Memory)
- Main Memory
- Program: Instructions and Data
Compiler and Interpreter

- Compiler
  - Source Code
    - (原始程式, t.c)
  - Object Code
    - (目的程式, t.obj)
  - Other Object Code
    - (Library)
  - Executable Code
    - (可執行檔, t.exe or t.com)
  - Computer

- Interpreter
  - Source Code
    - (原始程式, Lisp, Java)
  - Interpreter
  - Computer

Comparison

- Compiler
  - 已事先編譯，再執行過程中不必再次翻譯
  - 執行速度快，效率高
  - 大部分的程式
- Interpreter
  - 不需要事先編譯，操作手續簡便
  - 執行速度慢
  - 學習容易，執行過程中可隨時指出程式的錯誤
  - Java (可跨平台執行)

1.7 History of C

- C
  - Evolved by Ritchie from two previous programming languages, BCPL and B
  - Used to develop UNIX
  - Now, most operating systems written with C or C++
  - Hardware independent (portable)
  - By late 1970's C had evolved to "Traditional C"
- Standardization
  - Many slight variations of C existed, and were in compatible
  - Committee formed to create a "unambiguous, machine-independent" definition
  - Standard created in 1989, updated in 1999
<table>
<thead>
<tr>
<th>1.8 The C Standard Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>C programs consist of pieces/modules called functions</td>
</tr>
<tr>
<td>A programmer can create his own functions</td>
</tr>
<tr>
<td>Advantage: the programmer knows exactly how it works</td>
</tr>
<tr>
<td>Disadvantage: time consuming</td>
</tr>
<tr>
<td>Programmers will often use the C library functions</td>
</tr>
<tr>
<td>Use these as building blocks</td>
</tr>
<tr>
<td>Avoid re-inventing the wheel</td>
</tr>
<tr>
<td>If a premade function exists, generally best to use it rather than write your own</td>
</tr>
<tr>
<td>Library functions carefully written, efficient, and portable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.9 The Key Software Trend: Object Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
</tr>
<tr>
<td>Reusable software components that model items in the real world</td>
</tr>
<tr>
<td>Meaningful software units</td>
</tr>
<tr>
<td>Date objects, time objects, paycheck objects, invoice objects, audio objects, video objects, file objects, record objects, etc.</td>
</tr>
<tr>
<td>Any noun can be represented as an object</td>
</tr>
<tr>
<td>Very reusable</td>
</tr>
<tr>
<td>More understandable, better organized, and easier to maintain than procedural programming</td>
</tr>
<tr>
<td>Favor modularity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.10 C++ and C++ How to Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
</tr>
<tr>
<td>Superset of C developed by Bjarne Stroustrup at Bell Labs</td>
</tr>
<tr>
<td>“Spruces up” C, and provides object-oriented capabilities</td>
</tr>
<tr>
<td>Objects - reusable software components</td>
</tr>
<tr>
<td>Object-oriented design very powerful</td>
</tr>
<tr>
<td>10 to 100 fold increase in productivity</td>
</tr>
<tr>
<td>Dominant language in industry and university</td>
</tr>
<tr>
<td>Learning C++</td>
</tr>
<tr>
<td>Because C++ includes C, some feel it is best to master C, then learn C++</td>
</tr>
<tr>
<td>Starting in Chapter 15, we begin our introduction to C++</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.11 Java and Java How to Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java is used to</td>
</tr>
<tr>
<td>Create Web pages with dynamic and interactive content</td>
</tr>
<tr>
<td>Develop large-scale enterprise applications</td>
</tr>
<tr>
<td>Enhance the functionality of Web servers</td>
</tr>
<tr>
<td>Provide applications for consumer devices (such as cell phones, pagers and personal digital assistants)</td>
</tr>
<tr>
<td>Java How to Program</td>
</tr>
<tr>
<td>Closely followed the development of Java by Sun</td>
</tr>
<tr>
<td>Teaches first-year programming students the essentials of graphics, images, animation, audio, video, database, networking, multithreading and collaborative computing</td>
</tr>
</tbody>
</table>
1.12 Other High-level Languages

- A few other high-level languages have achieved broad acceptance
  - FORTRAN
    - Scientific and engineering applications
  - COBOL
    - Used to manipulate large amounts of data
  - Pascal
    - Intended for academic use

1.13 Structured Programming

- Structured programming
  - Disciplined approach to writing programs
    - Clear, easy to test and debug, and easy to modify
  - Multitasking
    - Specifying that many activities run in parallel

1.14 Basics of a Typical C Program Development Environment

- Phases of C++ Programs:
  1. Edit
  2. Preprocess
  3. Compile
  4. Link
  5. Load
  6. Execute

1.15 Hardware Trends

- Every year or two the following approximately double:
  - Amount of memory in which to execute programs
  - Amount of secondary storage (such as disk storage) to hold programs and data over the longer term
  - Processor speeds at which computers execute their programs
1.16 History of the Internet

- The Internet enables
  - Quick and easy communication via e-mail
  - International networking of computers
- Packet switching
  - Transfer digital data via small packets
  - Allows multiple users to send and receive data simultaneously
- No centralized control
  - If one part of the Internet fails, other parts can still operate
- Bandwidth
  - Information carrying capacity of communications lines

1.17 History of the World Wide Web

- World Wide Web
  - Locate and view multimedia-based documents on almost any subject
  - Makes information instantly and conveniently accessible worldwide
  - Possible for individuals and small businesses to get worldwide exposure
  - Changing the way business is done

1.18 General Notes About C and This Book

- Program clarity
  - Programs that are convoluted are difficult to read, understand, and modify
- C is a portable language
  - Programs can run on many different computers
  - However, portability is an elusive goal
- We will do a careful walkthrough of C
  - Some details and subtleties not covered
  - If you need additional technical details
    - Read the C standard document
    - Read the book by Kernigan and Ritchie