

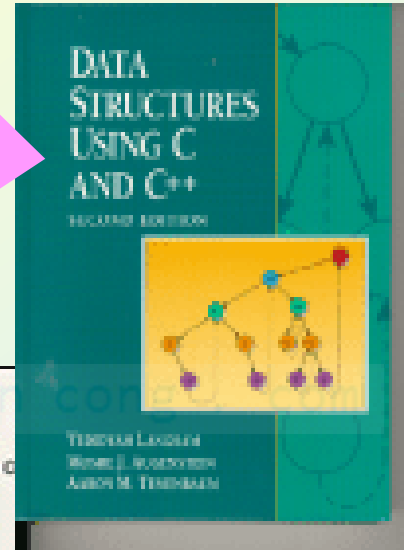
Data Structures and *Algorithms*

Reference : Dr. Helena Wong – www.cs.cityu.edu.hk

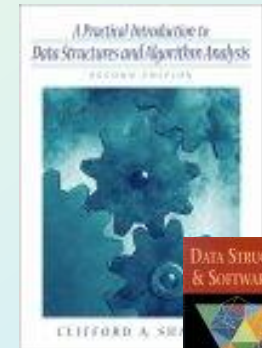
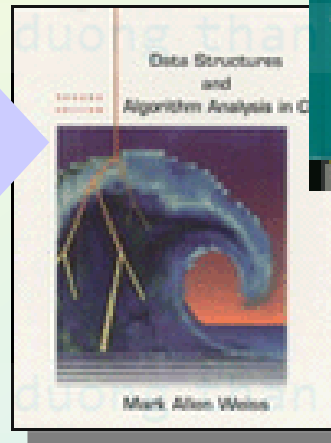
Lecturer : MSc. Trinh Quoc Son
University of Information Technology
Email : sontq@uit.edu.vn

Reference Books

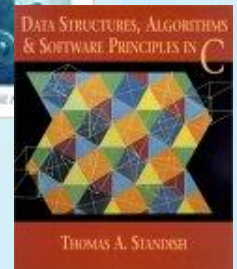
Data Structures Using C and C++
By Langsam, Augenstein, Tenenbaum
[Prentice Hall]



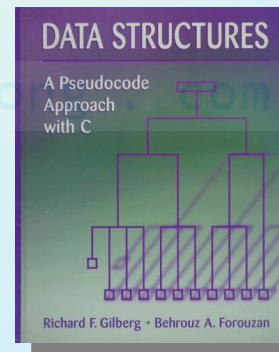
Data Structures and
Algorithm Analysis in C
By Mark Allen Weiss
[Addison Wesley]



Shaffer



Standish



Gilberg

Programming Language and Tools

- We will write programs in **C language**
- Assignment/exercises can be use with Visual C++

Week 1,2 (at home)

- Review C programming
- If you have missed something, find your tutor immediately!

If you consider other C programming tools:

eg. Visual Studio.Net, Pelles C (<http://www.smorgasbordet.com/pellestc/>)

- The compiler acts in the same way as Visual C++? **Mostly not.**

Indentation (use **tabs**) / format:

```
if (x>y)
county++;
else
if (x>z)
{countz++;
countx++;}
```



```
if (x>y)
→county++;
else
→if (x>z)
→{→countz++;
→countx++;
→}
```

GOOD

Hard coding should be avoided:

```
if (x >36)
{ x=0;
y++;
}
```



```
const int WIDTH=36;
..
if (x >WIDTH)
{ x=0;
y++;
}
```

GOOD

Using
named
constant

Global variables:

only if necessary and appropriate

```
int i, j;
void count()
{ for (i=0;...
} ...
void main()
{ for (i=0;...
} ...
```



```
const int SIZE=10;
int table[SIZESIZE];
void PrintTable()
{..
}
void main()
{..
}
```

OK

Comments (Enough but not excess!)

- Describe the program at the beginning
- Describe important variables and global declarations
- Describe each function before the function
- Explain any complex logic

Program bugs are normal -- You should fix them!

Design of logic ..
Coding ..
Testing ..
!! ERROR



Trust yourself:

Your program is based on a **workable logic**, right?

Now, the error is probably due to **careless mistakes** only .

With 99.9% sure these mistakes can be caught by using the **debugger**.



But after I have **corrected** the bug, the program still goes wrong. Should I undo the correction?

No. Don't undo the correction.

Now there is probably another small bug that you need to fix.

Don't give up. You are approaching to success.

Any problem in fixing a **program bug**?

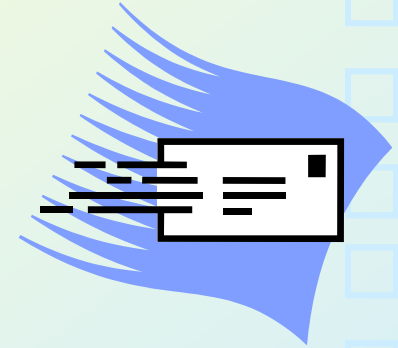
- First of all: solve for all compilation warnings (these are usually hints of bugs)
- A general debugging cycle:
 1. Test it again with **simple** test cases (increase complexity gradually)
 2. Select a **simplest** test case that your program runs incorrectly.
 3. With the test case, trace the code with pencil and paper.
(write down all intermediate results obtained step-by-step)
 4. Run your program with debugger to trace the program
(adding breakpoints / tracing line by line)
Locate the statement that causes any difference compared with your pencil-and-paper tracing.
 5. Fix the bug.
 6. Test the program with another test case.

Trust yourself on the logic: “the bug should be a careless mistake and can be fixed” – otherwise you will never start debugging.

get Help (MSDN): a convenient reference for C.

Any problem in this course?

- Send me **emails** to ask any question of this course.
- Email subject should be relevant eg. “help”.
- Attach any related program source code and test case.



I may contact you by email.

**If you prefer NOT to receive my email,
please inform me as soon as possible.**