

>>

COMP2521 ◇ Week 01a ◇ Course Introduction ◇ (20T3)

- People and Website
- Course Goals
- Course Context
- Revision (material from COMP1511)
- Data Structure Viewpoint
- COMP2521 Themes
- Lectures
- Tutes and Labs
- Assignments
- Plagiarism
- Final Exam
- Supplementary Exams
- Course Assessment
- C vs COMP1511 vs COMP2521
- Switch-statements
- For-loops

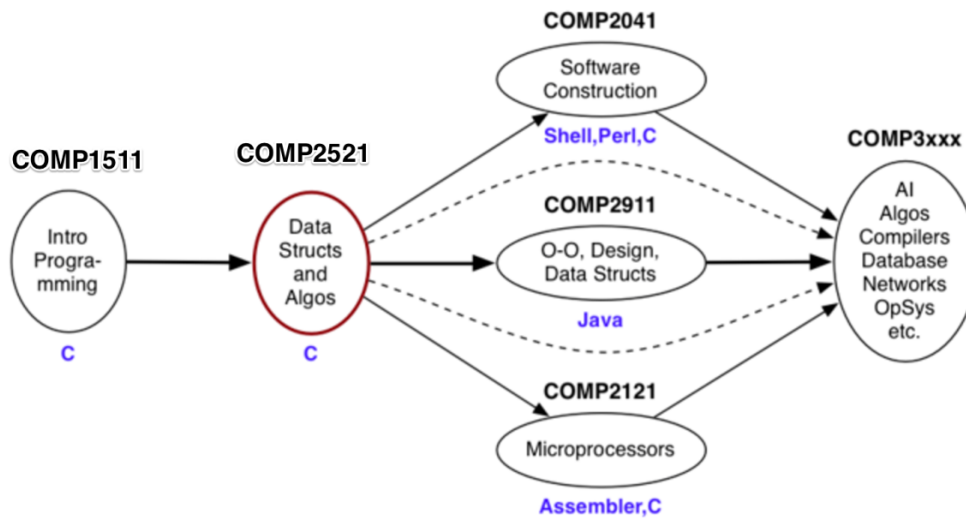
COMP2521 (20T3) ◇ Week 01a ◇ Course Intro ◇ [0/34]

❖ Thinking like a Scientist



观察 假设 实验 分析
observe → hypothesize → experiment → analyse → repeat

❖ Course Context



❖ Post-conditions

◇ At the *end* of this course you should be able to:

- analyse performance characteristics of algorithms
- measure performance behaviour of programs
- choose/develop effective data structures (DS)
- choose/develop algorithms (A) on these DS
- package a set of DS+A as an abstract data type
- develop and maintain 9999-line C programs

❖ Data Structure Viewpoint

COMP1511 considered ...

Atomic (char)

Atomic (int)

Atomic (float)

Linear (array)

0	1	2	3	4								
---	---	---	---	---	--	--	--	--	--	--	--	--

Tuple (struct)

--	--	--	--

Linear (list)



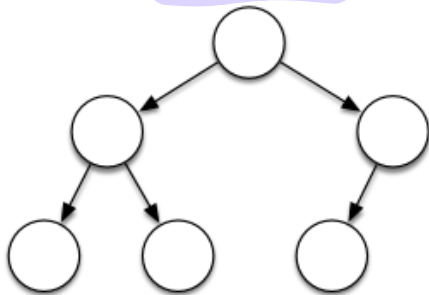
❖ ... Data Structure Viewpoint

COMP2521 also considers ...

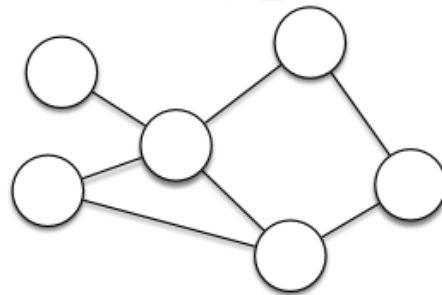
Linear (list)



Branching (tree)



Cyclic (graph)



❖ COMP2521 Themes

◇ Major themes ...

1. Analysis: correctness, performance, *usability*
2. ADTs: sets, lists, trees, graphs, dictionaries
3. Operations: building, sorting, searching, traversing

◇ **For data types:** alternative implementation of operations

◇ **For algorithms:** complexity analysis, performance analysis

❖ ... Tutes and Labs

- ◇ Lab exercises contribute 11% to overall mark.
- ◇ The lab exercises for Week X must be
 - submitted before Sunday at end of week X
 - demonstrated to tutor *during* Week X lab
OR, demonstrated *at the start of* Week X+1 lab
- ◇ Total mark for all the labs is greater than 11 (and they are scaled to 11).

❖ Assignments

◇ Two assignments ...

- Ass1: 15% towards final mark, (available in week-03, due Mon week-07)
- Ass2: 20% towards final mark, (available in week-07, due in week-10)

◇ Two assignments contribute 35% towards final mark.

◇ Late penalties apply if you miss the deadline.

◇ Good time management avoids late penalties!

tuor mark

COMP2521 (20T3) ◇ Week 01a ◇ Course Intro ◇ [18/34]

❖ Final Exam

- ◇ 3-hour on-line exam during the exam period.
- ◇ How to pass? Practice, practice, practice, ...

❖ COMP2521 Style

◇ Extends the range of allowed constructs:

- layout: consistent indentation still required
- use of brackets: { }
- can omit if control structure owns a single statement
- put function start bracket on line after function header
- can use all C control structures
- **if, switch, while, for, break, continue**
- put function start bracket on line after function header

◇ But wait! There's more ...

❖ ... COMP2521 Style

◇ More allowed C constructs ...

- can use assignment statements in expressions
 - but you should continue to avoid other kinds of side-effects
- can use conditional expressions
 - but use `x = c ? e1 : e2` with care
- functions may have multiple **return** statements
 - use very sparingly, primarily for error handling

```

if (x > 0) {
    y = x - 1;  z = x + 1;
} else {
    y = x + 1;  z = x - 1;
}
  
```

$y = (x > 0) ? x - 1 : x + 1;$
 $\Rightarrow z = (x > 0) ? x + 1 : x - 1;$

COMP2521 (20T3) ◇ Week 01a ◇ Course Intro ◇ [27/34]

❖ ... COMP2521 Style

◇ **Good:** gives you more freedom and power

- more choice in how you express programs
- can write code which is more concise (simpler)

◇ **Bad:** gives you more freedom and power

- can write code which is more cryptic 含义模糊
- can lead to 难理解 **incomprehensible, unmaintainable** code

◇ So, you **must** still use some discipline. 自律

❖ Switch-statements

◇ **switch** encapsulates a common selection:

```
if (v == C1) {  
    S1;  
} else if (v == C2) {  
    S2;  
}  
...  
else if (v == Cn) {  
    Sn;  
}  
else {  
    Sn+1;  
}
```

❖ ... Switch-statements

◇ Multi-way **if** becomes:

```
switch (v) {  
  case C1:  
    S1; break;  
  case C2:  
    S2; break;  
  ...  
  case Cn:  
    Sn; break;  
  default:  
    Sn+1;  
}
```

(v=C₁) 的情况下

默认

◇ Note: **break** is critical; if not present, falls through to next case.

❖ Exercise : Displaying Months

- ◇ Write a function `monthName(int)` that
- accepts a month number 1=Jan ... 12=Dec
 - returns a string containing the month name
 - assume that the string will be read-only
 - use a `switch` to decide on the month
- ◇ Suggest an alternative approach using an array.

❖ For-loops

◇ **for** encapsulates a common loop pattern:

```
init;  
while (cont) {  
    do something;  
    incr;  
}
```

as

```
for (init; cont; incr)  
    do something;
```

❖ ... For-loops

◇ COMP1511 (**while**) version

```
sum = 0;
i = 1;
while (i < 10) {
    sum = sum + i;
    i++;
}
```

◇ COMP2521 (**for**) version

```
sum = 0;
for (i = 0; i < 10; i++)
    sum += i;
```

= sum = sum + i;

❖ Exercise : Sequence program, using for

Write a program that prints integer sequences (one per line):

- `seqq 1-10` prints 1 2 3 4 5 6 7 8 9 10
- `seqq 5~10` prints 5 6 7 8 9 10
- `seqq 10~1` prints 10 9 8 7 6 5 4 3 2 1
- `seqq 1 3 10` prints 1 4 7 10 +3 +3 ...
- `seqq 1 3 11` prints 1 4 7 10
- `seqq 1~ -3` prints 1 0 -1 -2 -3
- `seqq 1 -3 10` gives an error

◇ Package the core part as a function:

```
void seqq(int start, int step, int finish) {...}
```

main checks errors and sets up args for **seqq()**

Produced: 13 Sep 2020