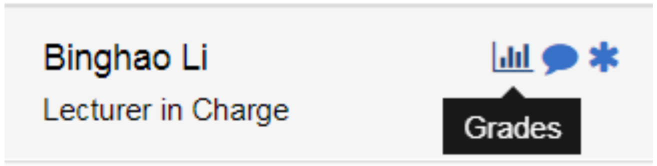
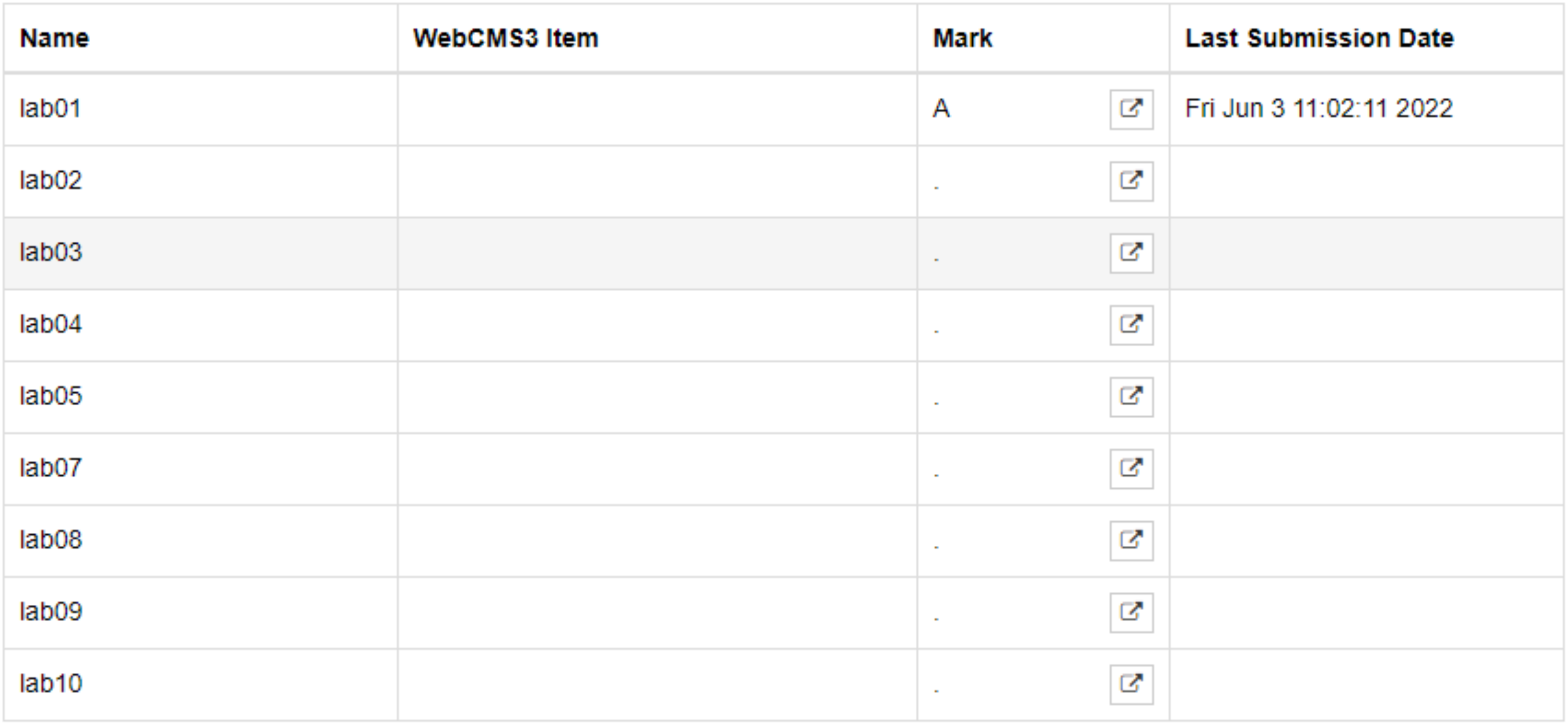
COMP1911 - Computing 1A

How to check if the assignment is submitted?

https://webcms3.cse.unsw.edu.au/COMP1911/23T2/


Binghao Li

Lecturer in Charge

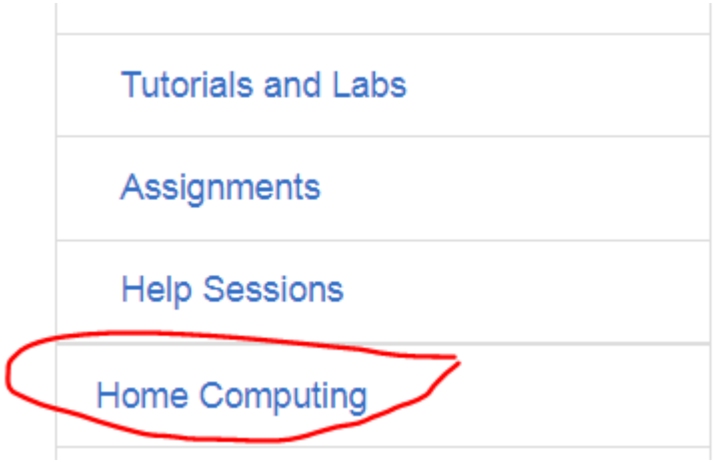


Name	WebCMS3 Item	Mark	Last Submission Date
lab01		A	Fri Jun 3 11:02:11 2022
lab02		.	
lab03		.	
lab04		.	
lab05		.	
lab07		.	
lab08		.	
lab09		.	
lab10		.	

Lab exercise

- No need to submit any tutorial work.
- Tutorial solutions will be released at Sunday 10pm at the end of the week of your tutorial.
- Lab exercises are due at Sunday 19:59:59，0.2% deduction per hour（5% per day).
- Lab solutions will be released at Sunday 10pm 2 weeks after they are due.

Home Computing



- How to use VLAB
- More information about connecting to the CSE machines
- If you are in overseas (especially in China), this link may be helpful: https://www.myit.unsw.edu.au/services/students/china-students-access-network
- You can have your own Linux system for exercises and assignments: Window Subsystem for Linux, Cygwin, Ubuntu …

Conversion specifier %lf or %f

```
double myMark = 52.567;
printf(“my mark is (1): %lf \n”, myMark);
printf(“my mark is (2): %6.2lf \n”, myMark);
printf(“my mark is (3): %10.2lf\n”, myMark);
printf(“my mark is (4): %2.2lf\n”, myMark);
```

4. C Conditions

In this lecture we will cover:

- More on Linux commands
- Making Choices
- Relational Operators
- Logical Operators
- If/else Statements

Navigating Unix

Linux commands are typed into a terminal.

To open a terminal on the default cse setup you can right click on the terminal icon at the bottom of the screen

Some useful linux commands to get started are:

- ls
- pwd
- mkdir
- cd

Linux Command: cp

- Linux Command **cp**: copies files and directories.
- cp *sourceFile destination*
- If the destination is an existing file, the file is overwritten
- If the destination is an existing directory the file is copied into the directory
- To copy a directory use cp -r *sourceDir destination*

Linux Command: mv

- Linux Command **mv** moves or renames a file.
- mv source destination
- If the destination is an existing file, the file is overwritten
- If the destination is an existing directory the file is moved into the directory.

Linux Command: rm

- Linux Command **rm** removes a file.
- Usually NO undo or recycle bin - be careful & have backups
- rm *filename*
- rm -r *directoryName*
  - ➢ This will delete a whole directory.
  - ➢ Be extra careful with this command

Control Flow

Problem: “read an integer and tell me if it's between 5 and 10.”

- We know how to read in an integer
- But how can we say whether it's less than 5?

What we need is a way of making choices in our programs. This functionality is known as control flow or branching and is provided by the if statement.

```c
int x;
scanf("%d", &x);
if (x > 5 && x < 10) {
  printf("Between 5 and 10!");
}
```

Before we can use if statements properly we need to understand relational operators and logical expressions.

When poll is active, respond at **pollev.com/binghaoli068**

Text **BINGHAOLI068** to **22333** once to join

Four types of brackets () [] {} <>, which one we have not touched yet?

- A ()
- B []
- C {}
- D <>
- None of the above

Total Results: 0

Powered by Poll Everywhere

Relational Operators

C has the usual operators to compare numbers:

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
!=	not equal to
==	equal to

- Don't confuse equality (==) with assignment (=)
- Be careful comparing doubles for equality using == or !=.
- Remember doubles are approximations.

Relational Operators

- Many languages have specific "boolean" types for TRUE and FALSE
- C does not have this type, so we just use int
- C convention is zero is false, other numbers true.
- All relational and logical operators return a "boolean":
  - the int **0** for false
  - the int **1** for true
- For example:

```
5 > 4   → 1
5 >= 4  → 1
5 < 4   → 0
5 <= 4  → 0
5 != 4  → 1
5 == 4  → 0
```

Logical Operators

Logical operators allow us to combine Boolean expressions (e.g., comparisons, etc.). We use them to answer questions like “Is x greater than y and less than z?”

The logical operators are:

and (&&)	true if both operands are true
or (||)	true if either operand is true
not (!)	true if its operand is false

Logical Operators

Truth tables show the results of logical operators with all different combinations of inputs

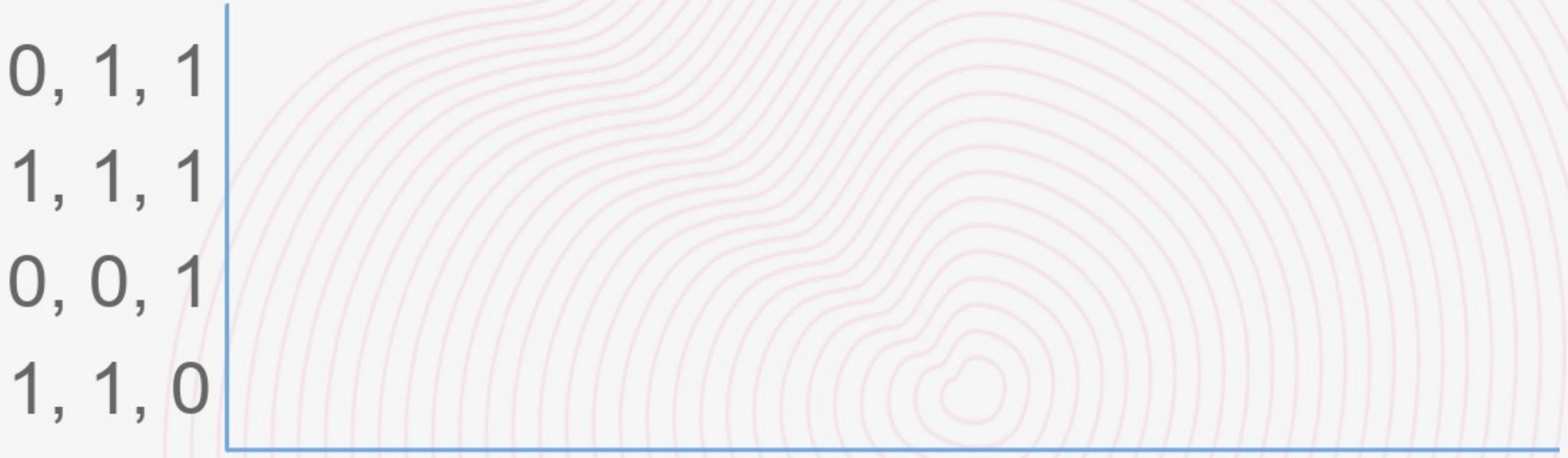
X	Y	X && Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X ||Y
0	0	0
0	1	1
1	0	1
1	1	1

&&, ||, !

Find the correct answers:

```
(2 > 0) && (2 < 2) ;
(0 > 1) || (2 < 10) ;
!(0 > 1)
```



Total Results: 0

Bitwise Operators

Operators	Meaning of operators
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise complement
<<	Shift left
>>	Shift right

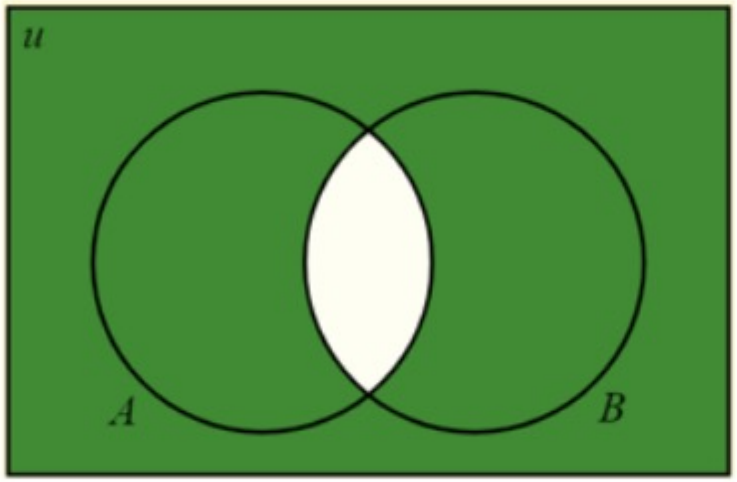
De Morgan's Laws

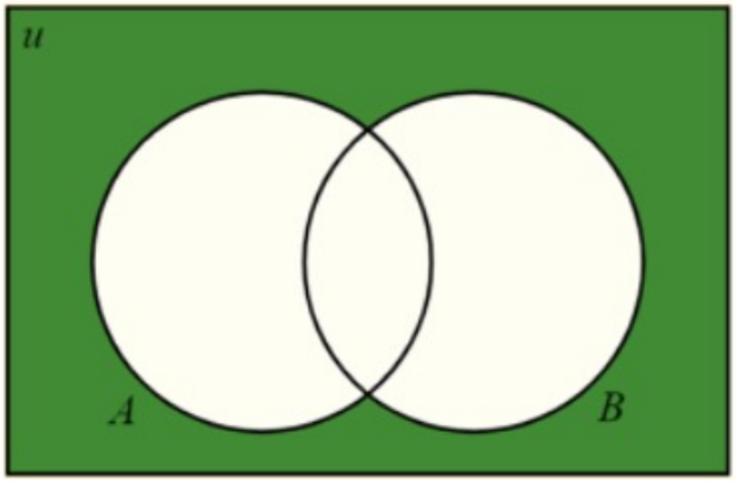
The rules can be expressed as:

- not (A or B) = not A and not B; and
- not (A and B) = not A or not B

Boolean algebra:

$$\overline{A \cup B} = \overline{A} \cap \overline{B},$$
$$\overline{A \cap B} = \overline{A} \cup \overline{B}.$$





Logical Operators / De Morgan's Laws

These two conditions are logically equivalent

```
!(height <= 130 && width <= 240)
```

.. is the same as ..

```
height > 130 || width > 240
```

Logical Operators / Short Circuit Evaluation

This is an important concept, the operators && and || evaluate their left-hand-side operand first and only evaluate their right-hand-side operand if necessary.

Operator && only evaluates its RHS if the **LHS is true**.
Operator || only evaluates its RHS if the **LHS is false.**

This is very useful because we can safely write:

```
(x != 0) && (y / x > 10)
```

Short Circuit Evaluation Examples

X=2 → (X>3 && X<10)

X>3 is False

No need to check X<10 as the LHS expression is false already, hence the compound expression evaluates to false

X=2 → (X>1 || X<10)

X>1 is True

No need to check X<10 as the LHS expression is true already, hence the compound expression evaluates to true

Short Circuit Evaluation:

Should we evaluate the second expression?

y=5; y>0 && x>100

y=5; y>4 || y<0.1

Y, N

N, Y

Y, Y

Total Results: 0

Precedence

A list of all operators in order of precedence, from high to low:

- !x, -x
- x * y, x / y, x % y
- x + y, x - y
- x < y, x <= y, x > y, x >= y
- x == y, x != y
- x && y (short-circuit left to right)
- x || y (short-circuit left to right)
- x = y

Explicit Order

The evaluation order can be changed and/or made explicit via parentheses, e.g., 7 * (4 + 3).

Don't Do This

Something like: 10 > x > 0 will compile (albeit with a compiler warning), but what does it mean? Suppose x = -1.

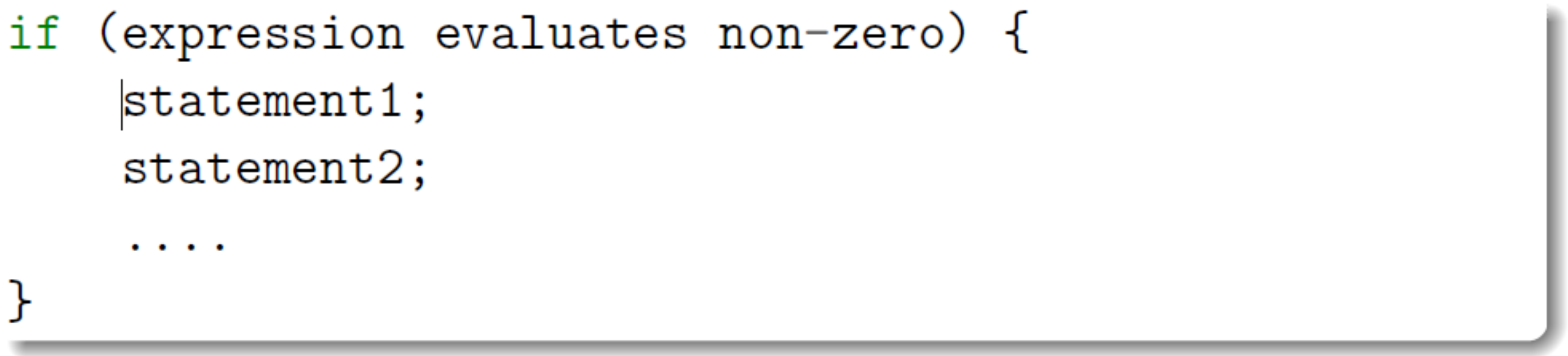
- ((10 > -1) > 0)
- (1 > 0)
- 1

What you probably mean to write is (10 > x) && (x > 0)

- (10 > -1) && (-1 > 0)
- 1 && (-1 > 0)
- 1 && 0
- 0

The if Statement

This is the structure of the if statement:



```
if (expression evaluates non-zero) {
    statement1;
    statement2;
    ....
}
```

- **statement1, statement2, ...** are executed if **expression** is non-zero.
- **statement1, statement2, ...** are NOT executed if **expression** is zero.

The else keyword

```
if (expression evaluates non-zero) {
    statement1;
    statement2;
    ....
} else if (expression evaluates non-zero) {
    statement3;
    statement4;
    ....
} else {
    statement5;
    statement6;
    ....
}
```

- **statement1, statement2** executed if **expression1** is non-zero.
- How are about statement3 - statement6?

The if Statement

We can also have nested if statements. i.e. if statements inside if statements

```
printf("%d is a ", a);
if (a < 0) {
    if (a < -100) {
        printf("big");
    } else if (a > -10){
        printf("small");
    } else {
        printf("medium");
    }
    printf(" negative");
} else {
    printf(" positive");
}
printf(" number.\n");
```

Things can be very complicated, be careful!

The if Statement

This syntax is also valid:

```
if (a == 0)
    printf("a is zero\n");
    a = 1; // this does not belong to if-block
```

If the braces ({}) are not supplied then the if statement controls only the statement that immediately follows.

Always use braces!

Doing this will ensure that you avoid bugs and ambiguity. The style guide requires it.

Questions