COMP20005 Workshop Week 3

1	**Q&A**: expressions, `if` statement
2	Discuss Exercise **`3.02`** Design **`3.06`**, 3.07. 5-min break
LAB	implement **`Ex 3.06`** using incremental development Design and implement a solution to **`Ex 3.07, 3.07b`**. [Extra] Other exercises from **`C03`**

if ...

```
if ( guard ) {
   // do some stuffs A1

}
// do B
```

```
if ( guard ) {
   // do some stuffs A1
} else {
   // do alternatives A2
}
// do B
```

Warming up: Revisit our equation program (in grok's Playground) *and try to improve it by*:

- solving the equation even if **a** is **0**
- learning to:
  - compile .c files, run executable files

More on `if` ...

Suppose that `a`, `b`, `c` were declared and had some value. Write a code segment to compute `min2`, `max3`, where:

- `min2` is the min of `a`, `b`
- `max3` is the max of `a`, `b`, `c`.

What can *`guard`* be?

if (guard)... : what can guard be

	our interpretation	C-machine's interpretation
guard	a condition that can be TRUE or FALSE	any expression non-zero value means TRUE, zero means FALSE
comparison like <, <=, >, >=, ==, !=	a comparison yields integer value TRUE or FALSE	a comparison yields integer value 1 or 0: - (1 is non-zero, meaning TRUE) - (0 means FALSE)
logical operations such as !, &&, ||	yields integer value TRUE or FALSE	yields integer value 1 or 0: - (1 is non-zero, meaning TRUE) - (0 means FALSE)
assignment a= b	a gets the current values of b	• a gets the current values of b • also a=b is an expression with value b

what can *guard* be? what's the operator precedence.

++, -- !, -, (*type*)	unary first!
*, /, % +, -	arithmetic
<, >, <=, >= ==, !=	relational
&& ||	logical
=, +=, *=, etc	assignments (right to left)

- Try Ex 3.01

(time limit: 5 minutes)

B a C Pro!

Compare:

	`a= 5;` `b= 5;`	`a= b= 5;` `Assignment is an expression!`

	`a= a * b;`
	`n= n+1;`
	`n += 1;`
	`m= m-1;`
		`a *= b;`
		`n++;`
		`m--;`

	`scanf("%d%d", &a, &b);`
	
	`//rest of the program`
		`if ( scantf("%d%d", &a, &b) != 2) {`
		`  printf("invalid input\n");`
		`  exit(EXIT_FAILURE);`
		`}`
		`// rest of the program`

Ex 3.02: use grok to *see and do* this exercise!

Trace the action of these statements, and determine the values printed out by each of the `printf` *statements. Assume that all variables have been declared to be of type* `int`.

Ex 3.2 a): (use grok and/or pens & papers)

```
1  i = 3;  j = 4;

2  if (  i<j  &&  j<6  ) {

3      i= i+j;
4  } else {
5      j= i+j;
6  }
7  printf ("i = %d, j = %d\n", i, j);
```

```
i = 7, j = 4
```

3.2 b)

```
1  i = 3;  j = 4;  k = 7;
2  if ((i<j || j<k) && j<i) {
3       i = i+1;
4       if (i*i>k) {
5            k = k+1;
6       }
7  } else {
8       j = j+1;
9       if (i*i>k) {
10           k = k+2;
11      }
12 }
13 printf ("i = %d, j = %d, k = %d\n", i, j, k);
```

```
i = , j = , k =
```

3.2 d)

```
1  x = 1; y = 2;
2  if (x>y)  {
3      printf (“x = %d, y = %d\n”, x, y);
4  }
5      x = x+1;
6
7  if (x<y) {
8  } ;


   {
```

x = , y =

3.2 e)

```
1   x = 1; y = 2;
2   if (x>y);
3   {
4       printf (“x = %d, y = %d\n”, x, y);
5       x = x+1;
6   }
7   if (x<y); {
8       printf (“x = %d, y = %d\n”, x, y);
9       y = y+2;
10  }
    printf (“x = %d, y = %d\n”, x, y);
```

```
x =  , y =
???
```

3.2 f)

```
1  x = 0; y = 0;
2  if (y<x) {
3      printf (“y is smaller\n”);
4  } else if (y=x) {
5      printf (“x and y are equal\n”);
6  } else {
7      printf (“y is greater\n”);
8  }
```

???

3.2 c)

```
1  month = 7;
2  if (month == 2) {
3      days = 28;
4  } else if (month == 4 || month==6 ||
5  month==9 || month==11) {
6      days = 30;
7  } else {
8      days = 31;
9  }
   printf (“days = %d\n”, days);
```

```
days =
```

Quiz 1

What **XXX** *should be in the following fragment:*

```
printf("Enter value for a and b : ");
if ( scanf("%d%d",&a,&b) XXX ) {
  printf("Please enter 2 integers\n");
  exit( EXIT_FAILURE );
}
```

A: `!= 0`	**B:** `!= 2`
C: `== 1`	**D:** `== 2`

Quiz 2

What is the output of the following fragment:

```
int a=1, b=2;
if ( a = b ) {
  printf("a= %d ", a);
} else {
  printf("b= %d", b);
}
printf("\n");
```

A: a= 1 b= 2	**B:** a= 1
C: a= 2	**D:** b= 2

How scan works? Quiz 3

If we execute the following fragment:

```
int i, n=0;  char c; float x;
n= scanf("%d%c%f", &i, &c, &x);
```

with the input:

```
100.1A200.2
```

the value of `n`, `i`, `c`, *and* `x` *become respectively:*

A: `0 100  A  200.2`	**B:** `3 100  A  200.2`
C: `3 100  .  1`	**D:** `(something else)`

`scanf("...", &a, &b)`

- tries to convert *as many as possible* input symbols to `a` according to the format
- if failed, it skips the whole `scanf` and does not consume input symbols
- if succeeded, it goes further and does the same for `b`
- returns the number of successful conversions

5-min break before LAB

Using `3.06` to discuss:

- **How to approach?**
- **What's the *benefit* of *incremental development*?**

`3.06`: *reads an integer amount of cents between 0 and 99 (check for valid input!) and print out the coins necessary to make up that amount of money.*

`3.07`: *Extend your "Fahrenheit to Celsius" program by adding in the reverse transformation.*

Then:

Ex3.01: Operator precedence
Ex3.03: Quadratic roots
Ex3.04: Date tomorrow
Ex3.05: Date with year
Ex3.07b: Generalised converter
Ex3.08: Calculating additional taxes

Additional Exercise:
Create A Quiz
(For those who have finished all today's lab works)

Write a program to perform your own quiz with around five questions. A question can require a number as an answer (e.g: what is the next number after 1 2 4 8?) or a selection (e.g. which choice (A, B, C, or D). After each question you should let the users know if they are correct.

And, at the end of the quiz, you should print the percentage of questions the user gets right. See example on the right.

```
Fun time!

What printf("%d\n", 5%2) prints out? 1
Correct!

Who is current Victoria's premier:
A Alistair   B Dan   C Jacinta   D Anh
? A
Incorrect

How many seasons in "Game of Thrones":
A 7       B  4       C  8     D  6
? C
Correct!

What is the output of:
if (0==1);printf("0=1 "); printf ("haha\n");
A  0=1 haha     B haha    C  <empty>
? haha
Incorrect

Not bad! You got 2 answers right.
Your score is 50%.
```

Wrap Up

operator precedence is as expected: *unary*, *arithmetic*, *relational*, *logic*, *assignment*

```
if ( guard ) { ... } else { ... }
```

guard : non-zero is TRUE, zero is FALSE

```
while ( guard ) {
   //do something in one iteration
}
```

reading one tuple of input

```
if (scanf(... ) != ...) {
    printf("...\n");
    exit(EXIT_FAILURE);
}
// process the input tuple
```

reading multiple tuples of input

```
while ( scanf(... ) == ... )
{
    // process next input tuple

}
```