ENG1008 Review 1
Topics 1- 4

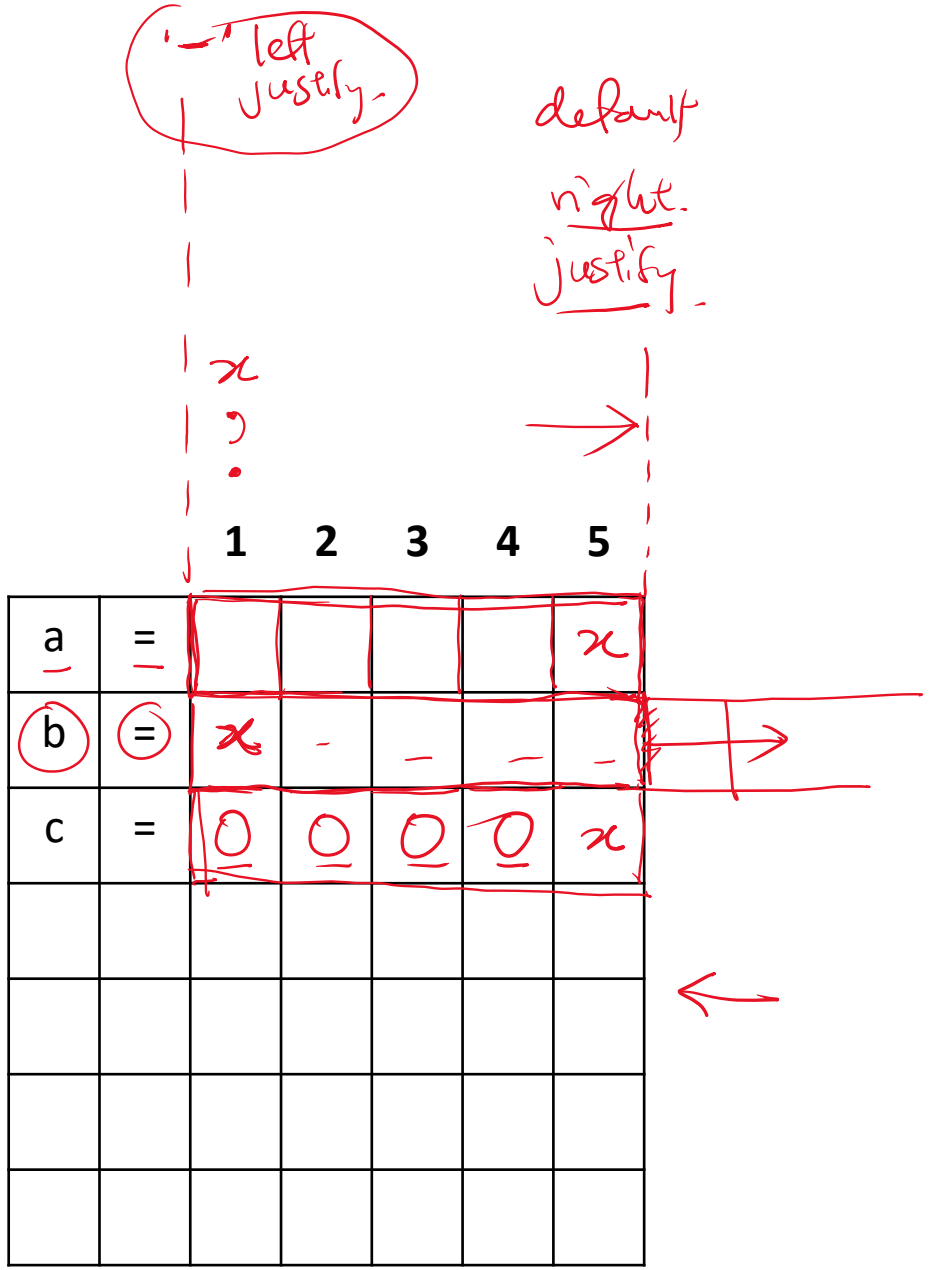
Figure 9.10 Format-Control-String Flags

Flag	Description
- (minus sign)	**Left justify** the output within the specified field.
+	Display a **plus sign** preceding positive values and a **minus sign** preceding negative values.
space	Print a space before a positive value not printed with the + flag.
#	Prefix 0 to the output value when used with the octal conversion specifier o. Prefix 0x or 0X to the output value when used with the hexadecimal conversion specifiers x or X. **Force a decimal point** for a floating-point number printed with e, E, f, g or G that does **not** contain a fractional part. (Normally the decimal point is printed **only** if a digit follows it.) For g and G specifiers, trailing zeros are not eliminated.
0 (zero)	Pad a field with **leading zeros.**

Example - Characters

State the values printed by the statements below.

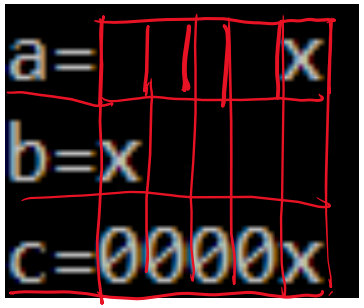
```
printf("a=%5c\n", 'x');
printf("b=%-5c\n", 'x');
printf("c=%05c\n", 'x');
```



Example - Characters

State the values printed by the statements below.

```
printf("a=%5c\n", 'x');
printf("b=%-5c\n", 'x');
printf("c=%05c\n", 'x');
```



		1	2	3	4	5
a	=					x
b	=	x				
c	=	0	0	0	0	X
		0000	x			

Example - String

%s

precision.

field width.

'x'

10 letters.

State the values printed by the statements below.

```
printf("a=%11s", "Marshmello");
printf("b=%-11s", "Marshmello");
printf("c=%11.5s", "Marshmello");
printf("d=%-11.5s", "Marshmello");
printf("e=%011.5s\n", "Marshmello");
```

two

1 2 3 4 5

pad with '0'

default right align.

		1	2	3	4	5	6	7	8	9	10	11
a	=	_	M	a	r	s	h	m	e	l	l	o
b	=	M	a	r	s	h	m	e	l	l	o	x
c	=							M	a	r	s	h
d	=	M	a	r	s	h						
e	=	0	0	0	0	0	0	M	a	r	s	h

Example - String

```
a= Marshmello
b=Marshmello
c=       Marsh
d=Marsh
e=000000Marsh
```

State the values printed by the statements below.

```
printf("a=%11s", "Marshmello");
printf("b=%-11s", "Marshmello");
printf("c=%11.5s", "Marshmello");
printf("d=%-11.5s", "Marshmello");
printf("e=%011.5s\n", "Marshmello");
```

		1	2	3	4	5	6	7	8	9	10	11
a	=		M	a	r	s	h	m	e	l	l	o
b	=	M	a	r	s	h	m	e	l	l	o	
c	=							M	a	r	s	h
d	=	M	a	r	s	h						
e	=	0	0	0	0	0	0	M	a	r	s	h

Example - integers

State the values printed by the statements below.

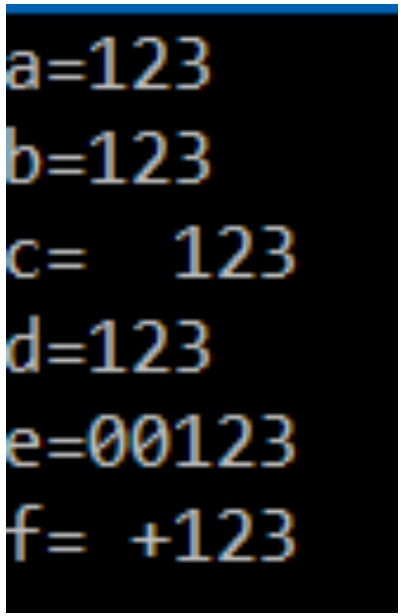
```
int x = 123;
printf("a=%d\n", x);
printf("b=%2d\n", x);
printf("c=%5d\n", x);
printf("d=%-5d\n", x);
printf("e=%05d\n", x);
printf("f=%+5d\n", x);
```

		1	2	3	4	5
a	=	1	2	3		
b	=	1	2	3		
c	=			1	2	3
d	=	1	2	3	x	x
e	=	0	0	1	2	3
f	=		+	1	2	3

Example - integers

State the values printed by the statements below.

```
int x = 123;
printf("a=%d\n", x);
printf("b=%2d\n", x);
printf("c=%5d\n", x);
printf("d=%-5d\n", x);
printf("e=%05d\n", x);
printf("f=%+5d\n", x);
```



		1	2	3	4	5
a	=	1	2	3		
b	=	1	2	3		
c	=			1	2	3
d	=	1	2	3		
e	=	0	0	1	2	3
f	=		+	1	2	3

Example - float

State the values printed by the statements below.

```
float x = 13.415035;
printf("a=%f\n", x);
printf("b=%8f\n", x);
printf("c=%10f\n", x);
printf("d=%010f\n", x);
printf("e=%8.2f\n", x);
printf("f=%08.2f\n", x);
printf("g=%-8.2f\n", x);
```

left align

```
a=13.415035
b=13.415035
c= 13.415035
d=013.415035
e=    13.42
f=00013.42
g=13.42
```

		1	2	3	4	5	6	7	8	9	10
a	=	1	3	.	4	1	5	0	3	5	
b	=	1	3	.	4	1	5	0	3	5	
c	=	_	1	3	.	4	1	5	0	3	5
d	=	0	1	3	.	4	1	5	0	3	5
e	=	_	_	_	1	3	.	4	2	r.a.	
f	=	0	0	0	1	3	.	4	2		
g	=	1	3	.	4	2					

%f

default right justify.

precision default = 6 dp

9 spaces needed.

& Spaces Specified

%8f

min width → auto expanded to meet spaces required.

%8.2f

13.42 — 2dp.

Back 3pm

Example - float

State the values printed by the statements below.

```
float x = 13.415035;
printf("a=%f\n", x);
printf("b=%8f\n", x);
printf("c=%10f\n", x);
printf("d=%010f\n", x);
printf("e=%8.2f\n", x);
printf("f=%08.2f\n", x);
printf("g=%-8.2f\n", x);
```

```
a=13.415035
b=13.415035
c= 13.415035
d=013.415035
e=   13.42
f=00013.42
g=13.42
```

		1	2	3	4	5	6	7	8	9	10
a	=	1	3	.	4	1	5	0	3	5	
b	=	1	3	.	4	1	5	0	3	5	
c	=		1	3	.	4	1	5	0	3	5
d	=	0	1	3	.	4	1	5	0	3	5
e	=				1	3	.	4	2		
f	=	0	0	0	1	3	.	4	2		
g	=	1	3	.	4	2					

TOH WEI QIANG BRYAN . to Everyone 2:48 PM

so prof,
you showed can convert Float to integer
is it possible to do the reverse?

Collapse All

Kua Zheng Hui 2:59 PM

can arh just put something like

```
#include <stdio.h>

main() {

int total = 17, values = 5;

float average;

average = (float) total / values;

printf("The average of all the values available with us is %f \n", average );

}
```

float → integer.

float x;
(int) x ;

int x
(float) x

integer division

(int) total / values = (17/5) = 3.4 (truncated) → 3

explicit type casting

average = 3.4

printf("%f", (float) total / value).

(float) total
(int) value

float implicit type cast

Example – wildcard for precision and field width

State the values printed by the statements below.

```
double x = 123456.789012345;
for (int fw=6, dp=0; fw <10; fw++, dp=(dp+1)%10) {
    printf("%0*.*lf\n", fw, dp, x);
}
```

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6				
1	2	3	4	5	6	.	8		
1	2	3	4	5	6	.	7	9	
1	2	3	4	5	6	.	7	8	9

Loop.	fw	dp	pmt. o/p
1	6	0	
2	7	1	
3	8	2	
4	9	3	
5	10	4	

Example – wildcard for precision and field width

State the values printed by the statements below.

```
double x = 123456.789012345;
for (int fw=6, dp=0; fw <10; fw++, dp=(dp+1)%10) {
    printf("%0*.*lf\n", fw, dp, x);
}
```

```
123457
123456.8
123456.79
123456.789
123456.7890
```

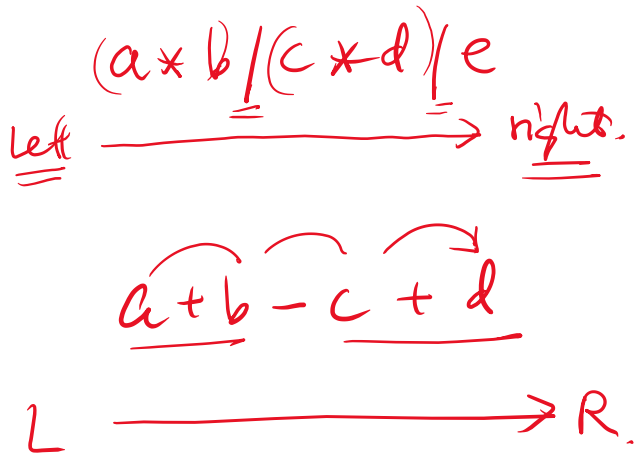
1	2	3	4	5	6	7	8	9	10		
1	2	3	4	5	7						
1	2	3	4	5	6	.	8				
1	2	3	4	5	6	.	7	9			
1	2	3	4	5	6	.	7	9	0		

Arithmetic in C

- **Parentheses** for grouping subexpressions
  - e.g. (a * b) / (c + d)
- **Rules of Operator Precedence**
  - similar rules to algebra
  - parentheses has the highest level of precedence

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the *innermost* pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they're evaluated left to right.
* / %	Multiplication Division Remainder	Evaluated second. If there are several, they're evaluated left to right.
+ -	Addition Subtraction	Evaluated third. If there are several, they're evaluated left to right.
=	Assignment	Evaluated last.

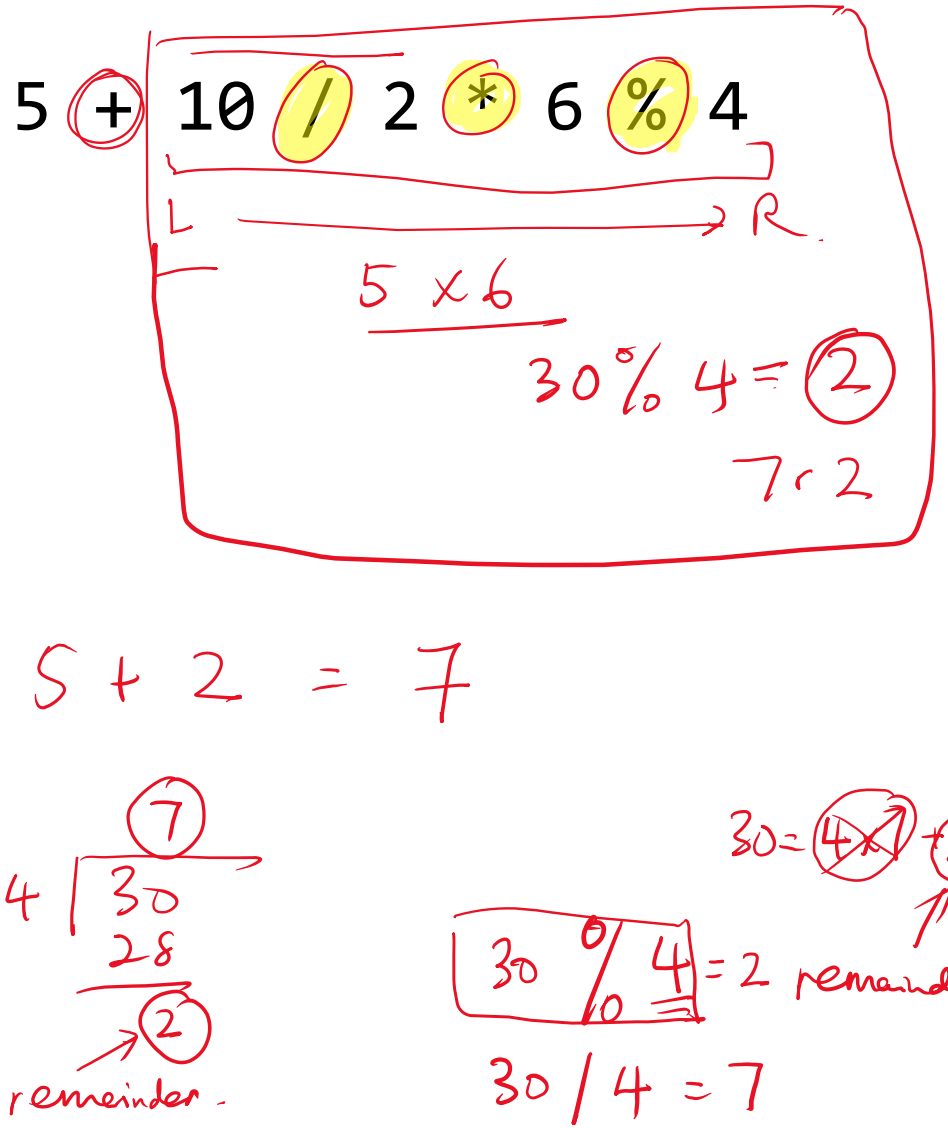
Fig. 2.10 | Precedence of arithmetic operators.



Example

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the *innermost* pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they're evaluated left to right.
*	Multiplication	Evaluated second. If there are several, they're evaluated left to right.
/	Division	
%	Remainder	
+	Addition	Evaluated third. If there are several, they're evaluated left to right.
-	Subtraction	
=	Assignment	Evaluated last.

Fig. 2.10 | Precedence of arithmetic operators.



Example

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the *innermost* pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they're evaluated left to right.
* / %	Multiplication Division Remainder	Evaluated second. If there are several, they're evaluated left to right.
+ -	Addition Subtraction	Evaluated third. If there are several, they're evaluated left to right.
=	Assignment	Evaluated last.

Fig. 2.10 | Precedence of arithmetic operators.

What is the value of z after executing the following code?

```
int x = 33;
double y = 5;
double z = 8;
y = x / 12;
z = 2 * y;
```

integer division

truncate / discard

33 ÷ 12 = 2

33/12 = 2

z = 2 x 2 = 4 ✓

y = 2.000000

3.50pm

Pseudocode

- Only contains **action statements** and the **program flow**
- Steps presented in a *structured* manner (numbered, indented, and so on)
- Emphasis is on *process*, not notation
- Well-understood forms allow *logical reasoning* about algorithm behavior
- *Definitions* (e.g. int i) are not executable statements
  - instructs the compiler to reserve memory space
  - does not cause any action
    - `int i` ← no action
    - `int i = 0` ← initialize variable i to zero

Several keywords are often used to indicate common input, output, and processing operations.

- Input: READ, OBTAIN, GET
- Output: PRINT, DISPLAY, SHOW
- Compute: COMPUTE, CALCULATE, DETERMINE
- Initialize: SET, INIT
- Add one: INCREMENT

Pseudocode

Step	Operation
1	*response* = Yes
2	While (*response* = Yes) do Steps 3 through 11
3	Get values for *gallons used*, *starting mileage*, *ending mileage*
4	Set value of *distance driven* to (*ending mileage* – *starting mileage*)
5	Set value of *average miles per gallon* to (*distance driven* ÷ *gallons used*)
6	Print the value of *average miles per gallon*
7	If average miles per gallon > 25.0 then
8	Print the message 'You are getting good gas mileage'
	Else
9	Print the message 'You are NOT getting good gas mileage'
10	Print the message 'Do you want to do this again? Enter Yes or No'
11	Get a new value for *response* from the user
12	Stop

Example

A C program is used to determine the result of a competition.

2 teams will participate in the competition and there are 10 tests. The results from the 10 tests are read and added to give the sum. The winning team will be the team with the highest sum.

The message “Draw” will be printed out if the sum for Team A is the same as Team B; “Team A is the winner” will be printed out if the sum for Team A is greater than Team B; “Team B is the winner” will be printed out if the sum for Team B is greater than Team A.

Develop the algorithm for this C program using pseudocode.

Example

A C program is used to determine the result of a competition.

2 teams will participate in the competition and there are 10 tests. The results from the 10 tests are read and added to give the sum. The winning team will be the team with the highest sum.

The message “Draw” will be printed out if the sum for Team A is the same as Team B; “Team A is the winner” will be printed out if the sum for Team A is greater than Team B; “Team B is the winner” will be printed out if the sum for Team B is greater than Team A.

Develop the algorithm for this C program using pseudocode.

BAD EXAMPLE

```
Declare sumA and sumB
Use a For loop to
   Read for Team A and B
   Calculate sumA and SumB
If sumA is equal to sumB
   Display “Draw”
Else If sumA is greater than sumB
   Display “Team A is the winner”
Else
   Display “Team B is the winner”
```

Example

A C program is used to determine the result of a competition.

2 teams will participate in the competition and there are 10 tests. The results from the 10 tests are read and added to give the sum. The winning team will be the team with the highest sum.

The message “Draw” will be printed out if the sum for Team A is the same as Team B; “Team A is the winner” will be printed out if the sum for Team A is greater than Team B; “Team B is the winner” will be printed out if the sum for Team B is greater than Team A.

Develop the algorithm for this C program using pseudocode.

```
Set sumA = 0
Set sumB = 0

For i = 1 to 10
   Read result for Team A
   Add result to sumA
   Read result for Team B
   Add result to sumB

If sumA is equal to sumB
   Display “Draw”
Else
   If sumA is greater than sumB
      Display “Team A is the winner”
   Else
      Display “Team B is the winner”
```

Example

A C program is used to determine the result of a competition.

2 teams will participate in the competition and there are 10 tests. The results from the 10 tests are read and added to give the sum. The winning team will be the team with the highest sum.

The message “Draw” will be printed out if the sum for Team A is the same as Team B; “Team A is the winner” will be printed out if the sum for Team A is greater than Team B; “Team B is the winner” will be printed out if the sum for Team B is greater than Team A.

Develop the algorithm for this C program using pseudocode.

```
Set sumA to 0
Set sumB to 0
Set i to 0

While i is smaller than 10
   Get result for Team A
   Add result to sumA
   Get result for Team B
   Add result to sumB
   Add 1 to i

If sumA is equal to sumB
   Display “Draw”
Else
   If sumA is greater than sumB
      Display “Team A is the winner”
   Else
      Display “Team B is the winner”
```