

COMP20005 Workshop Week 3

0	if statement and simple for loop
1	Discuss Exercise 3.2
2	Design 3.6
	5-min break
L A B	implement Exercise 3.6 using incremental development
	[Time permitting] Design and implement a solution to Exercise 3.7.

if . . .

```
if (a < b) {  
    printf("a is smaller than b\n");  
} else {  
    printf("a is NOT smaller than b\n");  
}
```

if (2>1) ...

if (a <=b && a <= c) ...

if (a) ...

if (100) ...

if (0) ...

fill in for computing the min of a,b,c

...

```
int main(int argc, char *argv[]) {  
    int a,b,c;  
    printf("Enter int value for a,b,c: ");  
    if (scanf("%d %d %d", &a, &b, &c) != 3) {  
        printf ("invalid input, I could not get 3 integers\n");  
        exit(EXIT_FAILURE);  
    }
```

...

```
    printf("The min of %d, %d, and %d is %d\n", a, b, c, ?);  
    return 0;  
}
```

Ex 3.2: use grok to see 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: see grok!

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`.

```
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 = , j =`

Ex 3.2: see grok!

Trace the action of these statements
printed out by each of the printf
variables have been declared

```
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);
```

What is?

the outcome of:

- relational operation such as <, <=
- logical operations &&, ||, !

i = , j =

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      x = x+1;  
5  if (x<y)  
6      printf ("x = %d, y = %d\n", x, y);  
7      y = y+2;  
8  printf ("x = %d, y = %d\n", x, y);
```

x = , y =

3.2 e)

```
1  x = 1; y = 2;
2  if (x>y); {
3      printf ("x = %d, y = %d\n", x, y);
4      x = x+1;
5  }
6  if (x<y); {
7      printf ("x = %d, y = %d\n", x, y);
8      y = y+2;
9  }
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 || 6 || 9 || 11) {  
5     days = 30;  
6 } else {  
7     days = 31;  
8 }  
9 printf ("days = %d\n", days);
```

days =

Operators

precedence

$++$, $--$

$!$, $-$, $(type)$

unary

$*$, $/$, $\%$

$+$, $-$

arithmetic

$<$, $>$, $<=$, $>=$

$==$, $!=$

relational

$\&\&$

$||$

logical

$=$, $+=$, $*=$, etc

assignments

example

1 && - 2 * - 3 - 4 < 5 && 6 <= 7 >= 8 != 9 / 10 > ! 11

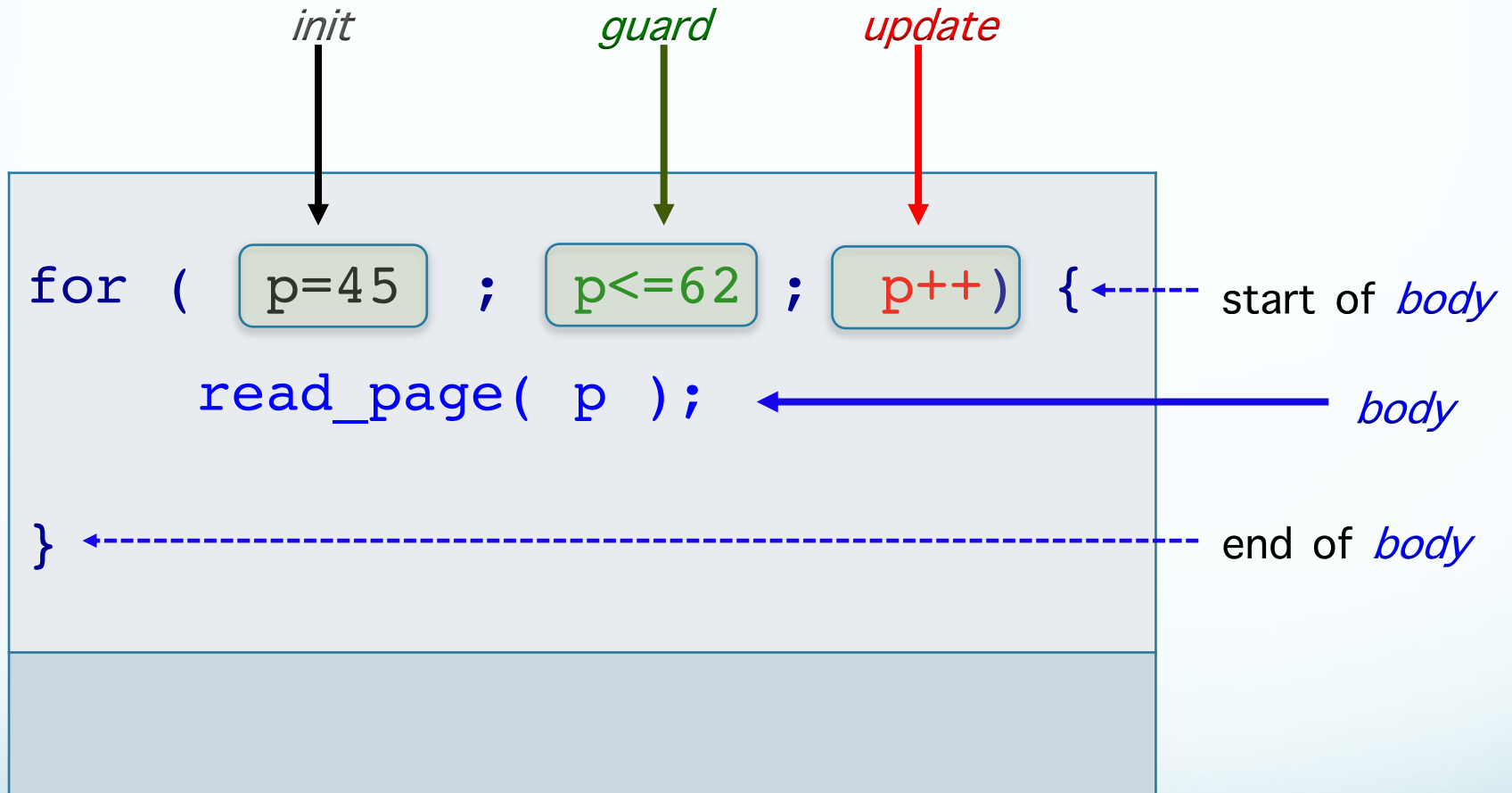
1 && - 2 * - 3 - 4 < 5 && 6 <= 7 >= 8 != 9 / 10 > ! 11

Loops

“program” for reading textbook chapter 4

1 2 3 17 18	<pre>read_page(45); read_page(46); read_page(47); ... read_page(61); read_page(62);</pre>	
note	<i>Pages 45-62 in the text book is chapter 4 (“Loops”).</i>	

Loops

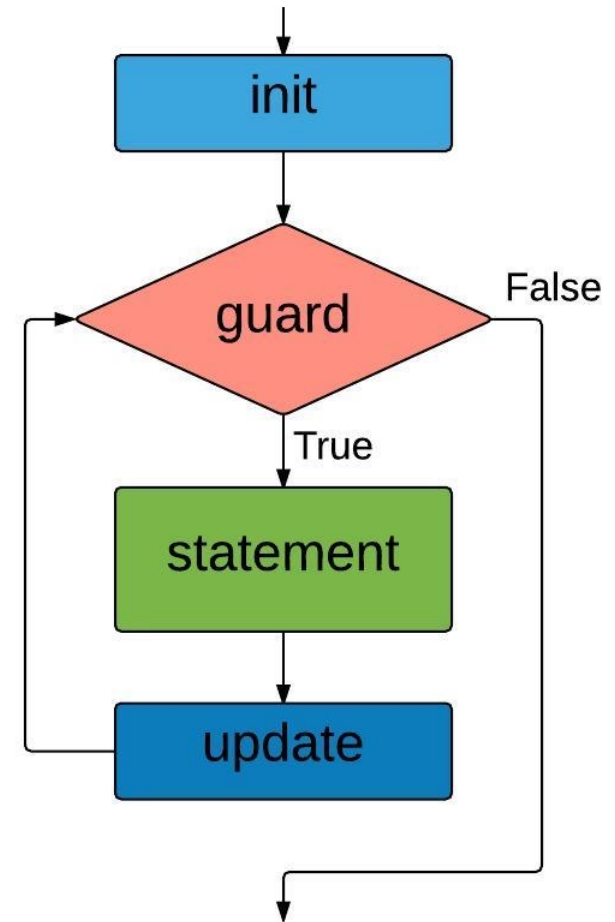


can be empty, empty means 1

```
for (init ; guard ; update) {  
    statement;  
}
```

All the boxes can be empty! The
must-be parts are:

```
for ( ; ; )
```



Examples: build code fragments for computing:

- $S = 1 + 2 + \dots + n$
- $S =$ sum of integers from input

B a C Pro!

Compare:

<pre>a= 5; b= 5;</pre>	<pre>a= b= 5; Assignment is an expression!</pre>
<pre>a= a * b; n= n+1; n += 1; m= m-1;</pre>	<pre>a *= b; n++; m--;</pre>
<pre>scanf("%d%d", &a, &b); //rest of the program</pre>	<pre>if (scanf("%d%d",&a,&b) != 2) { printf("invalid input\n"); exit(EXIT_FAILURE); } // rest of the program</pre>

Quiz 1

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 stream (data from keyboard) of:

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)

Quiz 2

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 3

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

5-min break

Ex 3.6 (Design)

In the past, Australia had coins in denominations of 50c, 20c, 10c, 5c, 2c, and 1c. Write a program that reads an integer amount of cents between 0 and 99 (your program might check for valid input) and print out the coins necessary to make up that amount of money. For example:

H:>calculatechange

Enter amount in cents: 41

The coins required to make 41 cents are:

give a 20c coin

give a 20c coin

give a 1c coin

amount remaining: 0c

Note: Don't worry if your program seems a bit clumsy, and not terribly general!

Ex 3.6 (Design) ?

Lab: Implement 3.6 and 3.7 and others in grok C03 and C04

3.6: *In the past, Australia had coins in denominations of 50c, 20c, 10c, 5c, 2c, and 1c. Write a program that reads an integer amount of cents between 0 and 99 (your program might check for valid input) and print out the coins necessary to make up that amount of money.*

3.7: *Extend your “Fahrenheit to Celsius” program by adding in the reverse transformation. For example:*

H:>converter

Enter a temperature: 212C

The temperature 212.0C converts to 413.6F

How about extending further for more units: M (miles), K (kilometers), P (Pound), G (kilogram)?

Finished?

Then do other exercises in grok C03/C04 and/or ask Anh for a funny exercise.

Remember

operators: outcomes, precedence order

guard: anything not zero is TRUE, only zero is FALSE

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

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

```
for (i=0; i<n; i++) {  
    ...  
}
```

a char data can be printed with %c or %d

```
#define EPSILON 1e-6
```

```
#define MYNAME "Mr Bean"
```

Additional Exercise: Create A Quiz

(For those who have finished all W3, W3X grok work)

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 user know if she/he is 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 Scott D Anh
? **A**
No

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**
No

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