

C Programming Language

Conditional statement

Dr. Mostafa A. Elhosseini

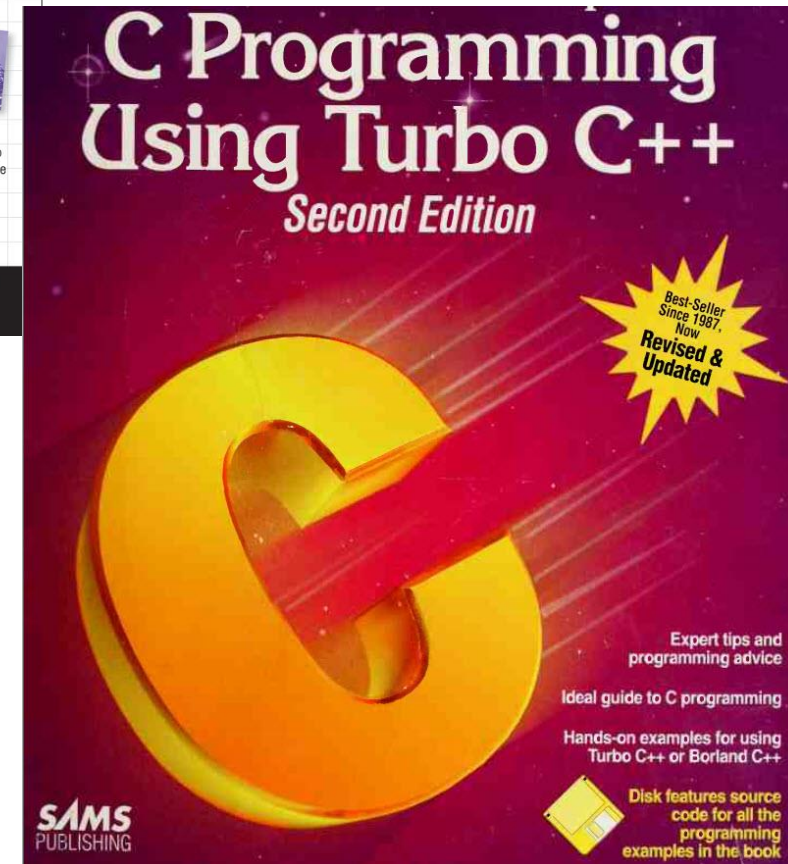
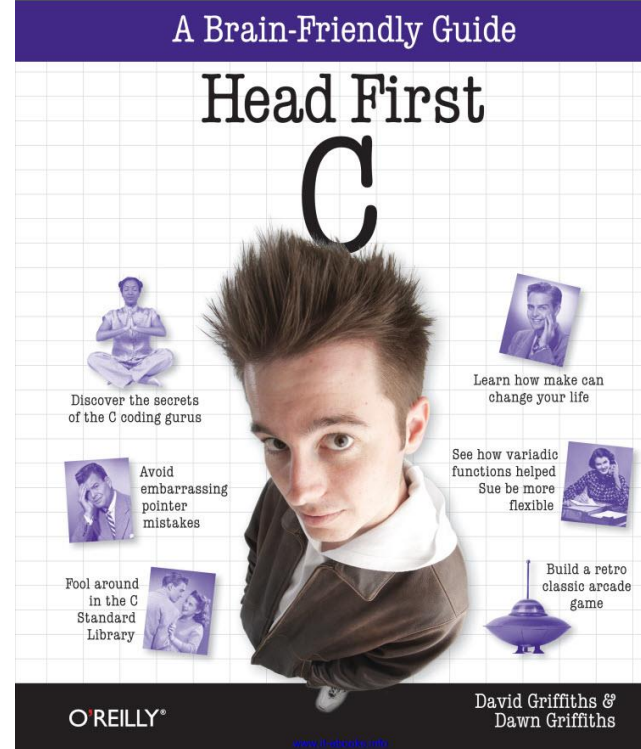
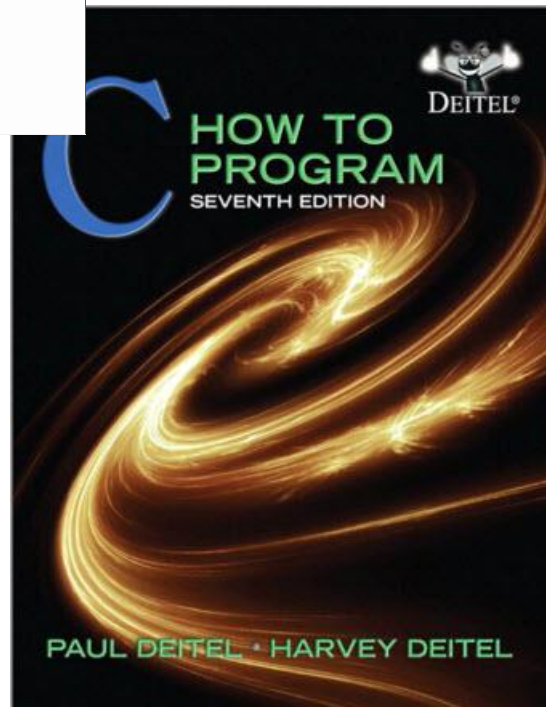
SECOND EDITION

THE



PROGRAMMING
LANGUAGE

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE



Revise

✂ Notes

- Comments
- **compiler** is only concerned with the types of the operands
- Casting
- Logical operators
- Bitwise operators

Agenda

- ✂ Conditional statement
 - if-statement

Algorithm

- ✂ A procedure for solving a problem in terms of
- the **actions** to be executed, and
 - the **order** in which these actions are to be executed
- is called an **algorithm**
- Get out of bed
 - Take a shower
 - Pray
 - Get dressed
 - Have breakfast
 - Go to work

Pseudocode

- ✧ **informal** language that helps you develop algorithms
- ✧ Pseudocode is **similar to everyday English**; it's convenient and user friendly although **it's not an actual computer programming** language
- ✧ Pseudocode consists only of action statements—those that are executed when the program has been converted from pseudocode to C and is run in C

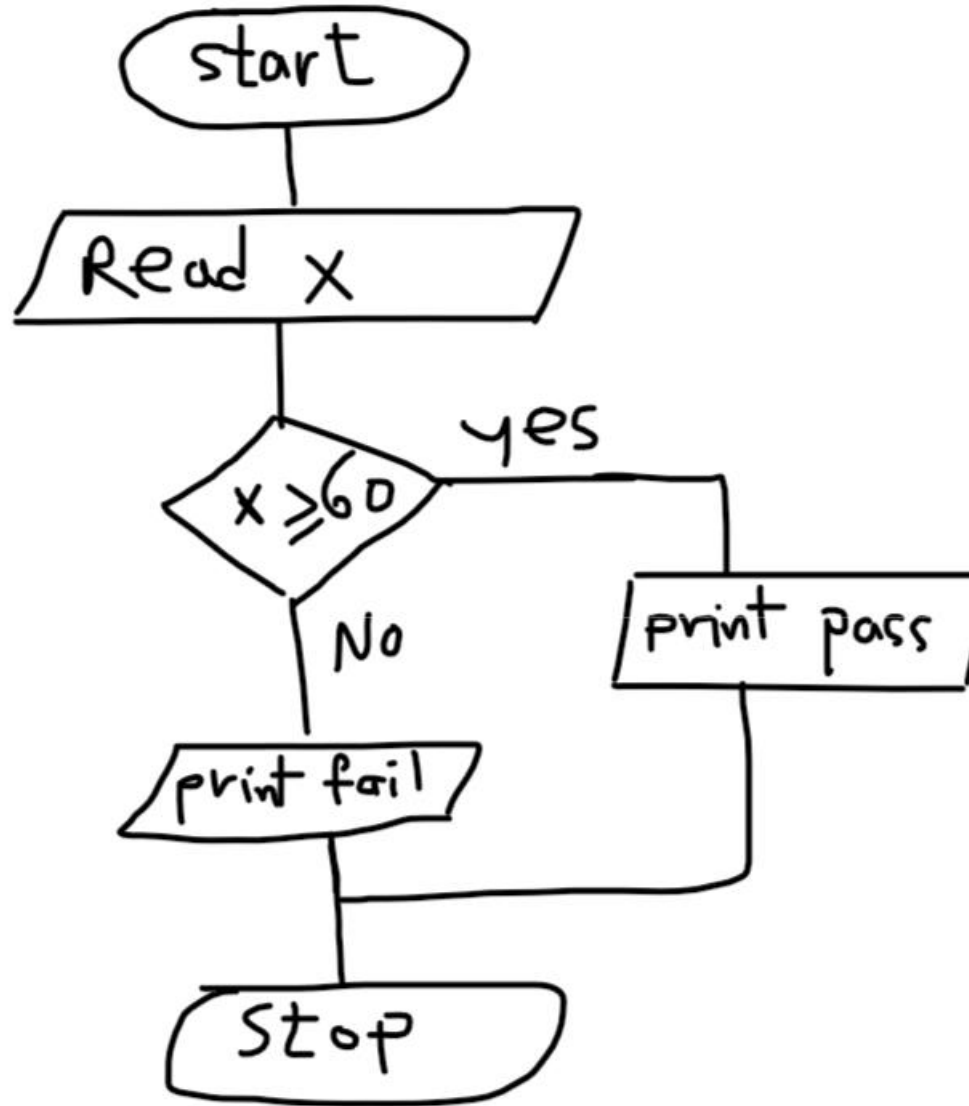
```
If student's grade is greater than or equal to 60  
    Print "Passed"
```

```
If student's grade is greater than or equal to 60  
    Print "Passed"  
else  
    Print "Failed"
```

Flowchart

- ✧ A **flowchart** is a **graphical representation** of an algorithm
- ✧ Flowcharts are drawn using certain **special-purpose symbols** such as **rectangles, diamonds, rounded rectangles**; these symbols are connected by **arrows**
- ✧ Like pseudocode, flowcharts are useful for developing and representing algorithms, although **pseudocode** is **preferred** by most programmers

■ Flowchart



Intro..

- ✧ We all need to be able to alter our actions in the face of changing circumstance.
 - If the **forecast** is for **rain**, then I will take my **raincoat**
 - If the **freeway** is under construction, then I will take the back road
- ✧ **Computer languages**, too, must be able to perform different sets of actions depending on the circumstances
- ✧ C has three major **decision-making** structures:
 - the **if** statement,
 - The **if -else** statement, and
 - the **switch** statement.

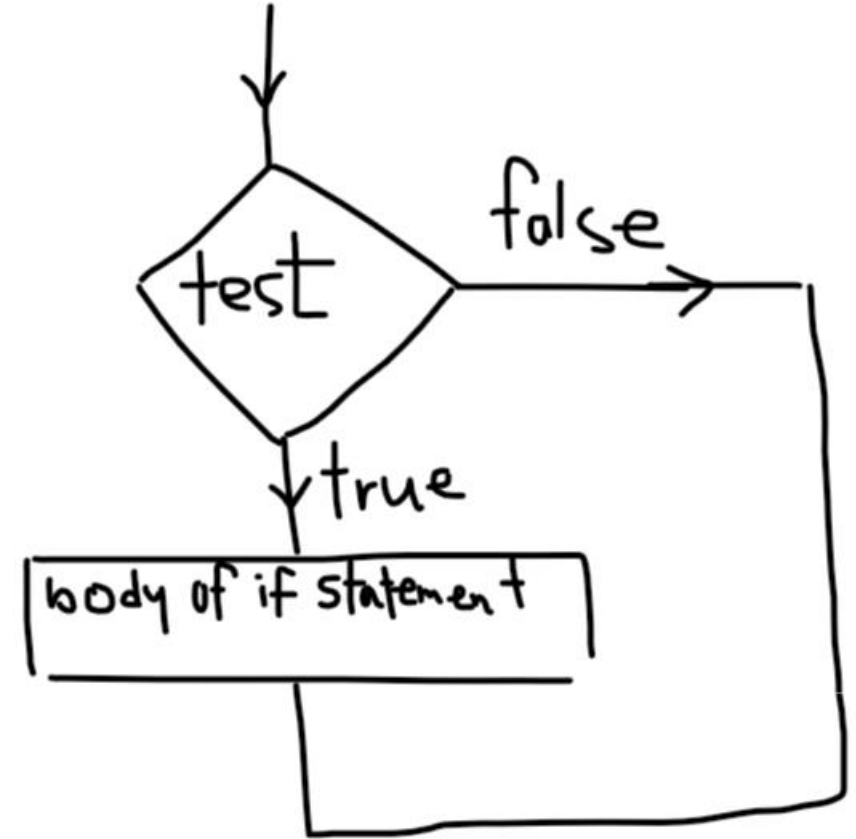
if statement

Keyword
if (x == 10)
{
printf("perfect");
}

Relational expression
True
false

body

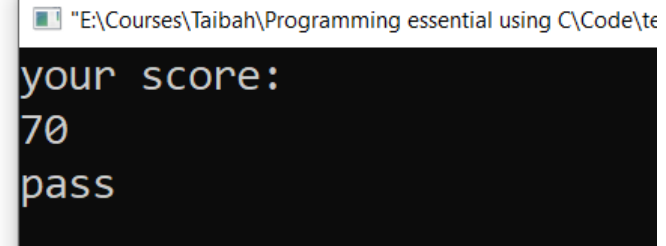
no semicolon here



■ if statement

- There is **no then** in if-statement
- Statements making up the **body** of the statement **won't be executed** at all if the **condition is false**.
- If the body of if-statement is **one-statement**, **no need to use the braces**
- **Indent** body statement of an **if**
- To **include several statements** in the body of an if, you **must enclose** the set of statements in **braces**

```
4  int main()  
5  {  
6      float x;  
7      puts("your score: ");  
8      scanf("%f", &x);  
9      if (x >= 60)  
10     {  
11         puts("pass");  
12     }  
13     return 0;  
14 }  
15
```



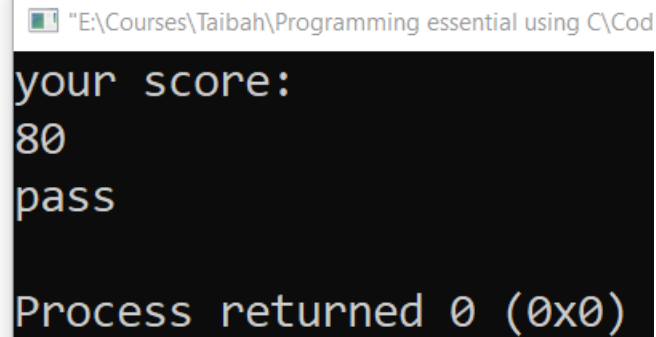
"E:\Courses\Taibah\Programming essential using C\Code\te
your score:
70
pass

A set of statements contained within a pair of braces is called a **compound statement** or a **block**

A **compound statement** can be **placed anywhere** in a program that a single statement can be placed

if statement

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      float x;
7      puts("your score: ");
8      scanf("%f", &x);
9      if (x >= 60)
10         puts("pass");
11     return 0;
12 }
```



```
"E:\Courses\Taibah\Programming essential using C\Cod
your score:
80
pass

Process returned 0 (0x0)
```

■ if statement

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      float x;
7      puts("your score:");
8      scanf("%f", &x);
9      if(x >= 60);
10     {
11         puts("pass");
12     }
13     return 0;
14 }
```

The compiler consider this placed for a reason and makes the semicolon the then part of the construct

"E:\Courses\Taibah\Programming essential using C\Code\tr

```
your score:
30
pass
```

Placing a *semicolon after* the condition in an if statement; leads to a *logic error* in single-selection if statements and a *syntax error* in double-selection if statements.

- Do no thing

```
4  int main()
5  {
6      float x;
7      puts("your score: ");
8      scanf("%f", &x);
9      if(x >= 60);
10     {
11         puts("pass");
12     }
13     else
14         puts("fail");
15     return 0;
16 }
17
```

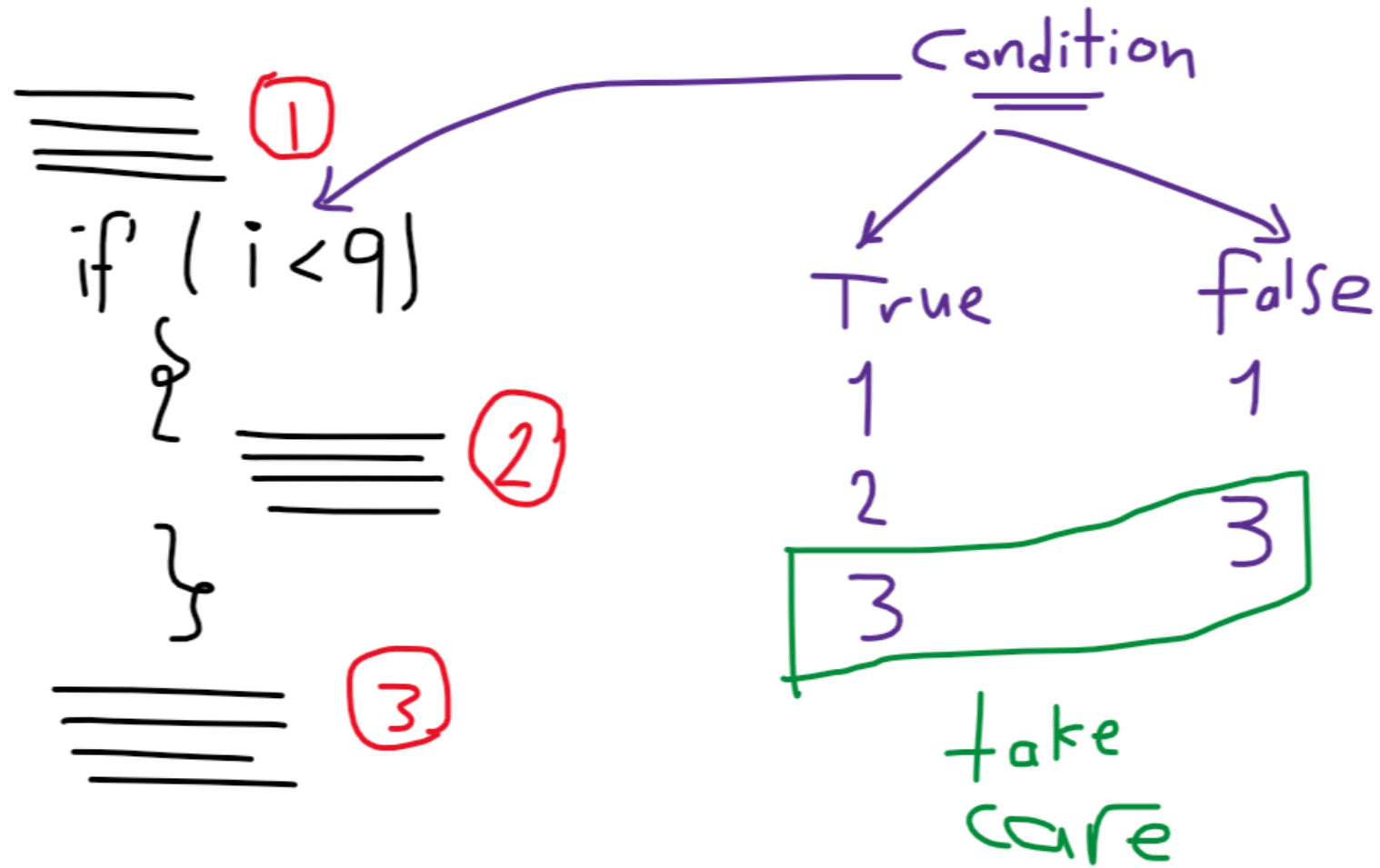
Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++

L.. Message

o... In function 'main':

o... 13 error: 'else' without a previous 'if'

■ if-statement



Example: Even or Odd

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int i;
6      puts("Your number: ");
7      scanf("%d", &i);
8      if(i%2 == 0)
9          printf("Even");
10     if(i%2 != 0)
11         printf("Odd");
12     return 0;
13 }
```