Nested if Lesson Outline

1. Nested if Lesson Outline
2. A Complicated if Example #1
3. A Complicated if Example #2
4. A Complicated if Example #3
5. A Complicated if Example #4
6. A Complicated if Example Run #1
7. A Complicated if Example Run #2
8. A Complicated if Example Run #3
9. Nested if Blocks
10. Nesting
11. Nested if Block Example #1
12. Nested if Block Example #2
13. How Nested if Blocks Work #1
14. How Nested if Blocks Work #2
15. Nested if Indentation
16. Nested if Block Example #1
17. Nested if Block Example #2
18. Nested if Block Example #3
19. Nested if Block Example #4
A Complicated if Example #1

```c
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    /* main */
    const int int_code = 1;
    const int float_code = 2;
    const int program_failure_code = -1;
    float float_input_value, float_output_value;
    int int_input_value, int_output_value;
    int data_type_code;
```
A Complicated if Example #2

printf("I’m going to calculate the");
printf(" absolute value\n");
printf(" of a number that you input.\n");
printf("Would you like to input ");
printf("an int or a float?\n");
printf(" (Enter %d for an int ", int_code);
printf("or %d for a float.)\n", float_code);
scanf("%d", &data_type_code);
if ((data_type_code != int_code) &&
    (data_type_code != float_code)) {
    printf("ERROR: I don’t recognize the ");
    printf("data type code %d.\n",
            data_type_code);
    exit(program_failure_code);
} /* if ((data_type_code != int_code) ... */
A Complicated if Example #3

```c
if (data_type_code == int_code) {
    printf("Please input the int value.\n");
    scanf("%d", &int_input_value);
    if (int_input_value < 0) {
        int_output_value = -int_input_value;
    } /* if (int_input_value < 0) */
    else {
        int_output_value = +int_input_value;
    } /* if (int_input_value < 0)...else */
    printf("The absolute value of ");
    printf("%d is %d.\n", int_input_value, int_output_value);
} /* if (data_type_code == int_code) */
```

Note that we’re using an int variable, data_type_code, to **encode** a quality rather than a quantity.
A Complicated if Example #4

```c
if (data_type_code == float_code) {
    printf("Please input the float value.\n");
    scanf("%f", &float_input_value);
    if (float_input_value < 0) {
        float_output_value = -float_input_value;
    } /* if (float_input_value < 0) */
    else {
        float_output_value = +float_input_value;
    } /* if (float_input_value < 0)...else */
    printf("The absolute value of ");
    printf("%f is %f.\n", float_input_value, float_output_value);
} /* if (data_type_code == float_code) */
} /* main */
```
A Complicated if Example Run #1

% gcc -o absvalbytype absvalbytype.c
% absvalbytype
I’m going to calculate the absolute value of a number that you input. Would you like to input an int or a float? (Enter 1 for an int or 2 for a float.)
0
ERROR: I don’t recognize the data type code 0.
A Complicated if Example Run #2

% absvalbytype
I’m going to calculate the absolute value of a number that you input. Would you like to input an int or a float? (Enter 1 for an int or 2 for a float.)

1
Please input the int value.

5
The absolute value of 5 is 5.

% absvalbytype
I’m going to calculate the absolute value of a number that you input. Would you like to input an int or a float? (Enter 1 for an int or 2 for a float.)

1
Please input the int value.

-5
The absolute value of -5 is 5.
A Complicated if Example Run #3

% absvalbytype
I’m going to calculate the absolute value of a number that you input. Would you like to input an int or a float? (Enter 1 for an int or 2 for a float.)

2
Please input the float value.

5.5
The absolute value of 5.500000 is 5.500000.

% absvalbytype
I’m going to calculate the absolute value of a number that you input. Would you like to input an int or a float? (Enter 1 for an int or 2 for a float.)

2
Please input the float value.

-5.5
The absolute value of -5.500000 is 5.500000.
Nested if Blocks

```c
if (condition) {
    if (condition) {
        statement;
        statement;
    } /* if (condition) */
else if (condition) {
    statement;
} /* if (condition) */
else {
    statement;
    statement;
} /* if (condition)...else */
} /* if (condition) */
else if (condition) {
    statement;
    if (condition) {
        statement;
        statement;
    } /* if (condition) */
    statement;
} /* if (condition) */
else if (condition) {
    statement;
} /* if (condition) */
else {
    if (condition) {
        statement;
    } /* if (condition) */
    else {
        statement;
        statement;
    } /* if (condition)...else */
} /* if (condition)...else */
```
Nesting

Nesting means putting something inside something else. For example, one if block can be nested inside another if block.

We refer to the inner if block as “the inner if block,” and we refer to the outer if block as “the outer if block.”

Go figure.
Nested if Block Example #1

#include <stdio.h>

int main ()
{ /* main */
   const int minimum_number = 1;
   const int maximum_number = 10;
   const int computers_number = 5;
   const int close_distance = 1;
   int users_number;

   printf("I’m thinking of a number between %d and %d.\n", minimum_number, maximum_number);
   printf("What number am I thinking of?\n");
   scanf("%d", &users_number);
Nested if Block Example #2

if ((users_number < minimum_number) ||
    (users_number > maximum_number)) {
    printf("Hey! That’s not between %d and %d!\n", 
           minimum_number, maximum_number);
} /* if ((users_number < minimum_number) || ... */
else if (users_number == computers_number) {
    printf("That’s amazing!\n");
} /* if (users_number == computers_number) */
else {
    printf("Well, at least you were within the range\n");
    if (abs(users_number - computers_number) <=
        close_distance) { 
        printf(" and you were close!\n");
    } /* if (abs(users_number-computers_number) <= ...) */
    else if (users_number < computers_number) {
        printf(" but you were way too low.\n");
    } /* if (users_number < computers_number) */
    else {
        printf(" but you were way too high.\n");
    } /* if (users_number < computers_number)...else */
    printf("My number was %d.\n", computers_number);
} /* if (users_number == computers_number)...else */
} /* main */
How Nested \texttt{if} Blocks Work #1

Suppose that an \texttt{if} block is nested inside another \texttt{if} block. What will happen?

Well, the sequence of statements \texttt{inside} a clause of an \texttt{if} block is executed only in the event that the clause’s condition evaluates to true (1) and all prior conditions within the \texttt{if} block evaluate to false (0) – or, in the case that the clause is an \texttt{else} clause, only in the event that all of the \texttt{if} block’s conditions evaluate to false (0).
**How Nested if Blocks Work #2**

On the other hand, an if statement is a normal executable statement (more or less).

So, in order for the inner if statement to be reached, and therefore executed, the outer clause that contains it must have a condition that evaluates to true (1), and all of the outer if block’s prior clauses must have conditions that evaluate to false (0) – or, in the case of an outer else clause, all of the conditions of the outer if block’s prior conditions must evaluate to false (0).

Once the inner if block is reached, it will be executed exactly like any other if block.
Nested if Indentation

Notice that the statements inside the nested if blocks are indented several extra spaces, so that it’s obvious which statements are inside which blocks.

In CS1313 programming projects, statements should be indented an extra four spaces for each block that they are inside.

We’ll see later that this rule applies not only to if blocks but to other kinds of blocks as well (for example, while loops).
Nested if Block Example #1

```c
#include <stdio.h>

int main ()
{ /* main */
    const int minimum_number = 1;
    const int maximum_number = 10;
    const int computers_number = 5;
    const int close_distance = 1;
    int users_number;
    printf("I’m thinking of a number between %d and %d.\n", minimum_number, maximum_number);
    printf("What number am I thinking of?\n");
    scanf("%d", &users_number);
}```
Nested if Block Example #2

```c
if ((users_number < minimum_number) ||
    (users_number > maximum_number)) {  
    printf("Hey! That's not between %d and %d!\n",  
           minimum_number, maximum_number);
} else if (users_number == computers_number) {  
    printf("That's amazing!\n");
} else {  
    printf("Well, at least you were within the range\n");  
    if (abs(users_number - computers_number) <=  
        close_distance) {  
        printf(" and you were close!\n");
    } else if (users_number < computers_number) {  
        printf(" but you were way too low.\n");
    } else {  
        printf(" but you were way too high.\n");
    }  
}  
printf("My number was %d.\n", computers_number);
```
Nested if Block Example #3

```c
% gcc -o nestedif nestedif.c
% nestedif
I’m thinking of a number between 1 and 10.
What number am I thinking of?
0
Hey! That’s not between 1 and 10!
% nestedif
I’m thinking of a number between 1 and 10.
What number am I thinking of?
11
Hey! That’s not between 1 and 10!
% nestedif
I’m thinking of a number between 1 and 10.
What number am I thinking of?
10
Well, at least you were within the range
but you were way too high.
My number was 5.
% nestedif
I’m thinking of a number between 1 and 10.
What number am I thinking of?
1
Well, at least you were within the range
but you were way too low.
My number was 5.
```
Nested if Block Example #4

\%
\textbf{nestedif}
I’m thinking of a number between 1 and 10.
What number am I thinking of?
4
Well, at least you were within the range
and you were close!
My number was 5.
\%
\textbf{nestedif}
I’m thinking of a number between 1 and 10.
What number am I thinking of?
6
Well, at least you were within the range
and you were close!
My number was 5.
\%
\textbf{nestedif}
I’m thinking of a number between 1 and 10.
What number am I thinking of?
5
That’s amazing!