if Lesson 3 Outline

1. if Lesson 3 Outline
2. Multiple, Related Conditions #1
3. Multiple, Related Conditions #2
4. Multiple, Related Conditions #3
5. else if Clause #1
6. if inside else
7. else if Clause #2
8. else if Clause #3
9. if-else if Can Short Circuit
10. if-else if Might Execute No Clause
11. if-else if Indenting
12. if-else if Clause Order
13. if-else if Example #1
14. if-else if Example #2
15. if-else if Example Flowchart
16. Mixing Branching Clauses #1
17. Mixing Branching Clauses #2
18. Mixing Branching Clauses #3
19. Mixing Branching Clauses #4
20. Mixing Branching Clauses #5
21. if - else if - else Example #1
22. if - else if - else Example #2
23. if - else if - else Flowchart
24. Multiple else if Clauses #1
25. Multiple else if Clauses #2
26. Multiple else if Clauses #3
27. Multiple else if Clauses #4
28. General Rule for Multiple else if Clauses
29. Order of Condition Evaluations
30. Multiple else if Example #1
31. Multiple else if Example #2
32. Multiple else if Flowchart
33. if, Multiple else if, else Clauses #1
34. if, Multiple else if, else Clauses #2
35. if, Multiple else if, else Clauses #3
36. if, Multiple else if, else Clauses #4
37. if, Multiple else if, else Clauses #5
38. if, Multiple else if, else Example #1
39. if, Multiple else if, else Example #2
Multiple, Related Conditions #1

What if we have multiple, related conditions and we want to be able to handle each?

Well, we could simply use multiple `if` blocks:

```c
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) || ... */
if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

That’s not too cumbersome.
Multiple, Related Conditions #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) || ... */
if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

**BUT**: Notice that there’s a case where **both** printf statements might be executed: in the event that **both**:

- `users_number` is less than `minimum_number`, and
- `users_number` is within `close_distance` of `computers_number`.

For example, if `minimum_number` were 1, `close_distance` were 2 and `users_number` were 3.
Multiple, Related Conditions #3

```c
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) || ... */
if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

Consider the case that **both**:
- `users_number` is less than `minimum_number`, and
- `users_number` is within `close_distance` of `computers_number`.

In that case, **both** outputs will be printed, which is **not** what we want; we want either to be told that we’re outside the range, or to be told that we’re close, but **not both**.
This approach looks okay, but you can imagine how pushed to the right the indenting would get if we had many of these in the same if block.
else if Clause #1

```c
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 (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

C allows us to set up another special clause of statements attached to the first `if` clause, called an `else if` clause.
else if Clause #2

if ((users_number < minimum_number) ||
   (users_number > maximum_number)) {
   printf("Hey! That's not between %d and %d!
", minimum_number, maximum_number);
} /* if ((users_number < minimum_number) || ... */
else if (abs(users_number - computers_number) <=
         close_distance) {
   printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

The statements inside the if clause are executed only in the event that the condition in the if statement evaluates to true (1).
else if Clause #3

```c
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 (abs(users_number - computers_number) <=
         close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

The statements inside the `else if` clause are executed only in the event that **both** of the following occur:

1. The `if` condition evaluates to false (0), **and**
2. the `else if` condition evaluates to true (1).
if-else if  Can Short Circuit

```
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 (abs(users_number - computers_number) <=
        close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

In the case that the `if` condition evaluates to true (1), it’s also the case that the `else if` condition isn’t evaluated at all. Why? Because in that case the statements inside the `else if` clause will be skipped regardless of the value of the `else if` condition, so the evaluation of the `else if` condition would be irrelevant. Why do work that isn’t going to help? This is another instance of `short circuiting`. 
if-else if Might Execute No Clause

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 (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

Notice that it could be the case that **no** clause of this if block
gets executed, in the event that **both** conditions evaluate to
false (0).
if-else if Indenting

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 (abs(users_number - computers_number) <=
         close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

The exact same indenting rules that apply to if clauses and else clauses also apply to else if clauses.
if-else if Clause Order

```c
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 (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

The order of the clauses is:

1. if clause (which **MUST** be **FIRST**), followed by
2. else if clauses, if any.
#include <stdio.h>

int main ()
{ /* main */
    const int computers_number = 5;
    int users_number;

    printf("Pick an integer:\n");
    scanf("%d", &users_number);
    if (users_number < computers_number) {
        printf("That’s unbelievable! Your number is\n");
        printf(" less than mine!\n");
        printf("Well, okay, maybe it’s believable.\n");
    } /* if (users_number < computers_number) */
    else if (users_number > computers_number) {
        printf("Surprise, surprise! Your number is\n");
        printf(" greater than mine\n");
    } /* if (users_number > computers_number) */
    printf("And now I’m sick of you.\n");
    printf("Bye!\n");
} /* main */
if-else if Example #2

% gcc -o islesselseif islesselseif.c
% islesselseif
Pick an integer:
  6
Surprise, surprise! Your number is greater than mine!
And now I’m sick of you.
Bye!
% islesselseif
Pick an integer:
  5
And now I’m sick of you.
Bye!
% islesselseif
Pick an integer:
  4
That’s unbelievable! Your number is less than mine!
Well, okay, maybe it’s believable.
And now I’m sick of you.
Bye!
printf("Pick an integer:\n");
scanf("%d", &users_number);
if (users_number < computers_number) {
    printf("That's unbelievable! Your number is");
    printf(" less than mine!\n");
    printf("Well, okay, maybe it's believable.\n");
} /* if (users_number < computers_number) */
else if (users_number > computers_number) {
    printf("Surprise, surprise! Your number is");
    printf(" greater than mine\n");
} /* if (users_number > computers_number) */
printf("And now I'm sick of you.\n");
printf("Bye!\n");
Mixing Branching Clauses #1

```c
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("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */
```

Here we have if, elseif and else clauses.

The statements inside the else clause are executed only in the event that **BOTH** the if condition **AND** the elseif condition evaluate to false (0).
Mixing Branching Clauses #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);
} /* if ((users_number < minimum_number) || ... */
else if (users_number == computers_number) {
    printf("That’s amazing!\n");
} /* if (users_number == computers_number) */
else {
    printf("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */
```

The presence of the `else` clause guarantees that at **exactly one** of the clauses of this `if` block will be executed.

If the `else` clause were absent, then it might be that no clause is executed, in the event that both of the conditions evaluated to false (0).
Mixing Branching Clauses #3

```c
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("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */
```

Again, notice that each clause has its own block open and its own block close.
Mixing Branching Clauses #4

```c
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("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */
```

Notice that the indenting rules that apply to `if` clauses and `else` clauses also apply to `else if` clauses.
Mixing Branching Clauses #5

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("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */

Notice that the if clause **MUST** be the **FIRST** clause of
the if block.

Notice that, **IF** there is an else clause, then
the else clause **MUST** be the **LAST** clause of
the if block.
#include <stdio.h>

int main ()
{ /* main */
    const int computers_number = 5;
    int users_number;

    printf("Pick an integer:\n");
    scanf("%d", &users_number);
    if (users_number < computers_number) {
        printf("That’s unbelievable! Your number is");
        printf(" less than mine!\n");
        printf("Well, okay, maybe it’s believable.\n");
    } /* (users_number < computers_number) */
    else if (users_number > computers_number) {
        printf("Surprise, surprise! Your number is");
        printf(" greater than mine!\n");
    } /* if (users_number > computers_number) */
    else {
        printf("Yowza! Your number is equal to mine!\n");
    } /* if (users_number > computers_number)...else */
    printf("And now I’m sick of you.\n");
    printf("Bye!\n");
} /* main */
if – else if – else

Example #2

```bash
% gcc -o islesselseifelse islesselseifelse.c
% islesselseifelse
Pick an integer:
6
Surprise, surprise! Your number is greater than mine!
And now I’m sick of you.
Bye!
% islesselseifelse
Pick an integer:
5
Yowza! Your number is equal to mine!
And now I’m sick of you.
Bye!
% islesselseifelse
Pick an integer:
4
That’s unbelievable! Your number is less than mine!
Well, okay, maybe it’s believable.
And now I’m sick of you.
Bye!
```
if

Lesson 3

CS1313 Fall 2020

if - else if - else

Flowchart

printf("Pick an integer:\n");
scanf("%d", &users_number);
if (users_number < computers_number) {
    printf("That’s unbelievable! Your number is\n");
    printf("Well, okay, maybe it’s believable.\n");
} /* (users_number < computers_number) */
else if (users_number > computers_number) {
    printf("Surprise, surprise! Your number is\n");
    printf("greater than mine!\n");
} /* if (users_number > computers_number) */
else {
    printf("Yowza! Your number is equal to mine!\n");
} /* if (users_number > computers_number)...else */
printf("And now I’m sick of you.\n");
printf("Bye!\n");

if - else if - else

Prompt for user’s number.

Input user’s number.

user’s < computer’s?

user’s > computer’s?

False

False

True

True

Output less.

Output greater.

Output equal.

Output goodbye.
Multiple `else if` Clauses #1

```c
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 if (abs(users_number - computers_number) <=
         close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

We don’t have to stop at just one `else if` clause;
we can have as many as we like.
```
if ((users_number < minimum_number) ||
    (users_number > maximum_number)) {
    printf("Hey! That’s not between %d and %d!
", 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 if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

As usual, the statements inside the if clause are executed
only in the event that the if condition (a Boolean
expression completely enclosed in parentheses) in
the if statement evaluates to true (1).
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 if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */

As usual, the statements inside the first else if clause are executed only in the event that both of two things occur:
1. The if condition evaluates to false (0), and
2. the first else if condition evaluates to true (1).
Multiple else if Clauses #4

```c
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 if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
```

The statements inside the 2\textsuperscript{nd} else if clause are executed only in the event that all of these things occur:

1. The if condition evaluates to false (0), and
2. the 1\textsuperscript{st} else if condition evaluates to false (0).
3. the 2\textsuperscript{nd} else if condition evaluates to true (1).
General Rule for Multiple else if Clauses

For a given else if clause, the statements inside it are executed only in the event that all of the following occur:

1. The if condition evaluates to false (0), and all prior else if conditions within the entire if block (in the event that there are any) evaluate to false (0), and
2. the given else if condition evaluates to true (1).
Order of Condition Evaluations

The conditions (Boolean expressions completely enclosed in parentheses) in the `if` statement and in the `elseif` statement(s) are evaluated until one of them results in true (1). Once a condition evaluates to true, then the conditions in all subsequent `elseif` statements within that `if` block are skipped.
#include <stdio.h>

int main ()
{ /* main */
    const int computers_number = 5;
    int users_number;

    printf("Pick an integer:\n");
    scanf("%d", &users_number);
    if (users_number < computers_number) {
        printf("That’s unbelievable! Your number is\n");
        printf(" less than mine!\n");
        printf("Well, okay, maybe it’s believable.\n");
    } /* (users_number < computers_number) */
    else if (users_number > computers_number) {
        printf("Surprise, surprise! Your number is\n");
        printf(" greater than mine!\n");
    } /* if (users_number > computers_number) */
    else if (users_number == computers_number) {
        printf("Yowza! Your number is equal to mine!\n");
    } /* if (users_number == computers_number) */
    printf("And now I’m sick of you.\n");
    printf("Bye!\n");
} /* main */
Multiple else if Example #2

% gcc -o islesselseifs islesselseifs.c
% islesselseifs
Pick an integer:
6
Surprise, surprise! Your number is greater than mine!
And now I’m sick of you.
Bye!
% islesselseifs
Pick an integer:
5
Yowza! Your number is equal to mine!
And now I’m sick of you.
Bye!
% islesselseifs
Pick an integer:
4
That’s unbelievable! Your number is less than mine!
Well, okay, maybe it’s believable.
And now I’m sick of you.
Bye!
printf("Pick an integer:\n");
scanf("%d", &users_number);
if (users_number < computers_number) {
    printf("That's unbelievable! Your number is\n");
    printf(" less than mine!\n");
    printf("Well, okay, maybe it's believable.\n");
} /* (users_number < computers_number) */
else if (users_number > computers_number) {
    printf("Surprise, surprise! Your number is\n");
    printf(" greater than mine!\n");
} /* if (users_number > computers_number) */
else if (users_number == computers_number) {
    printf("Yowza! Your number is equal to mine!\n");
} /* if (users_number == computers_number) */
printf("And now I'm sick of you.\n");
printf("Bye!\n");
if, Multiple else if, else Clauses #1

```c
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 if (abs(users_number - computers_number) <=
    close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
else {
    printf("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */
```

Not surprisingly, we not only can have as many else if clauses as we like, we can also have an else clause as well, as the LAST clause of the entire if block.
The statements inside the `else` clause are executed only in the event that the `if` condition and all of the `else if` conditions evaluate to false (0).
if, Multiple else if, else Clauses #3

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 if (abs(users_number - computers_number) <=
        close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
else {
    printf("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */

Notice that the statements inside the else clause will be
executed only in the events that all of the conditions within
the entire if block evaluate to false (0).
If an `if` block has an else clause, then the presence of that `else` clause guarantees that **exactly one** of the clauses of the `if` block will be executed.
if, Multiple else if, else Clauses #5

```c
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 if (abs(users_number - computers_number) <=
       close_distance) {
    printf("Close, but no cigar.\n");
} /* if (abs(users_number - computers_number) <= ... */
else {
    printf("Bzzzt! Not even close.\n");
} /* if (users_number == computers_number)...else */
```

Each clause has its own block open and block close.


# Multiple if, else if, else Example #1

```c
#include <stdio.h>

int main ()
{
    const int computers_number = 5;
    int users_number;

    printf("Pick an integer:\n");
    scanf("%d", &users_number);

    if (users_number < computers_number) {
        printf("That’s unbelievable! Your number is\n");
        printf(" less than mine!\n");
        printf("Well, okay, maybe it’s believable.\n");
    } /* if (users_number < computers_number) */
    else if (users_number > computers_number) {
        printf("Surprise, surprise! Your number is\n");
        printf(" greater than mine!\n");
    } /* if (users_number > computers_number) */
    else if (users_number == computers_number) {
        printf("Yowza! Your number is equal to mine!\n");
    } /* if (users_number == computers_number) */
    else {
        printf("This statement will never be executed.\n");
        printf(" Why?\n");
    } /* if (users_number == computers_number)...else */
    printf("And now I’m sick of you.\n");
    printf("Bye!\n");
} /* main */
```

if Lesson 3
CS1313 Fall 2020
38
% gcc -o islesselseifseiselse islesselseifseifseiselse.c
% islesselseifseifseiselse
Pick an integer:
6
Surprise, surprise! Your number is greater than mine!
And now I’m sick of you.
Bye!
% islesselseifseifseiselse
Pick an integer:
5
Yowza! Your number is equal to mine!
And now I’m sick of you.
Bye!
% islesselseifseifseiselse
Pick an integer:
4
That’s unbelievable! Your number is less than mine!
Well, okay, maybe it’s believable.
And now I’m sick of you.
Bye!