

CS 1313 010: Programming for Non-majors, Spring 2025

Commenting `if` Blocks

You ARE allowed to use the code examples in this document.

Commenting compound statements such as `if` blocks is somewhat more complicated than commenting simple statements such as `printf` statements.

(1) In the event that the `if` block has only an `if` clause, and no other clauses, then this is an example of the appropriate format:

```
/*
 * Idiotproof the side dish item code.
 */
if ((side_dish_item_code != no_item_code)    &&
    (side_dish_item_code != fries_item_code) &&
    (side_dish_item_code != rice_item_code)  &&
    (side_dish_item_code != onion_rings_item_code)) {
/*
 * Output the idiotproof error message for
 * the side dish item code.
 */
    printf("ERROR: unknown side dish item code %d.\n",
           side_dish_item_code);
/*
 * Exit the program because the side dish item code is
 * invalid.
 */
    exit(program_failure_code);
} /* if ((side_dish_item_code != no_item_code) && ...) */
```

WE URGE YOU TO USE THE CODE JUST ABOVE IN YOUR PP#4!

Notice:

- The comment preceding the entire `if` block describes the purpose of the `if` block as a whole.
- The block close of the `if` block is immediately followed, on the same line, by:
 - a single space, followed by
 - the comment open delimiter, followed by
 - a single space, followed by
 - the entire `if` statement, **EXCLUDING** the block open, followed by
 - the comment close delimiter.
- The comment on the same line as the block close is **ABSOLUTELY FORBIDDEN** to continue on to the next line.
- In the event that the `if` condition is very long, then in the comment on the associated block close line, you can use the first part of the condition, followed by an ellipsis (three periods).
- The comments for the statements inside the `if` clause are indented farther, so that the asterisk at the beginning of each line of the comment lines up with the statement that the comment describes.
- There is no comment just before each block close, because block closes are not statements.

(2) In the event that the `if` block has both an `if` clause and an `else` clause, but no others:

```
/*
 * Idiotproof the side dish item code.
 */
if ((side_dish_item_code != no_item_code)    &&
    (side_dish_item_code != fries_item_code) &&
    (side_dish_item_code != rice_item_code)  &&
    (side_dish_item_code != onion_rings_item_code)) {
    /*
     * Output the idiotproof error message for
     * the side dish item code.
     */
    printf("ERROR: unknown side dish item code %d.\n",
           side_dish_item_code);
    /*
     * Exit the program because the side dish item code is
     * invalid.
     */
    exit(program_failure_code);
} /* if ((side_dish_item_code != no_item_code) && ...) */
else {
    printf("This isn't a real example, it's just to illustrate.\n");
} /* if ((side_dish_item_code != no_item_code) && ...)...else */
```

DON'T USE THE CODE JUST ABOVE IN YOUR PP#4! (Or at least not its `else` clause.)

Notice that, in addition to the properties of just an `if` clause, above:

- The comment preceding the entire `if` block describes the purpose of the `if` block as a whole, not just the purpose of the `if` clause.
- There is no comment immediately before the `else` statement.
- The `else` statement is immediately followed, on the same line, only by a blank space and then the block open.
- The block close for the `else` statement is immediately followed, on the same line, by:
 - a single space, followed by
 - the comment open delimiter, followed by
 - a single space, followed by
 - the entire `if` statement, **EXCLUDING** the block open, followed by
 - three periods (that is, an ellipsis), followed by
 - the keyword `else`, to indicate that this is the end of the `else` clause, followed by
 - the comment close delimiter.
- Again, the comment on the same line as the block close is **ABSOLUTELY FORBIDDEN** to continue on to the next line.
- Again, in the event that the `if` condition is very long, then in the comment on the associated block close line of the `else` clause (that is, the final block close of that `if` block), you can use the first part of the condition, followed by an ellipsis (three periods).
- Again, the comments for the statements inside the `else` clause are indented farther, so that the asterisk at the beginning of each line of the comment lines up with the statement that the comment describes.
- Again, there is no comment immediately preceding each block close, because block closes are not statements.

(3) In the event that the `if` block has both an `if` clause and one or more `else if` clauses, but no `else` clause:

```
/*
 * Output the side dish name and price.
 */
if (side_dish_item_code == fries_item_code) {
    /*
     * Output the fries side dish name and price.
     */
    printf("    Fries:                %5.2f\n", side_dish_price);
} /* if (side_dish_item_code == fries_item_code) */
else if (side_dish_item_code == rice_item_code) {
    /*
     * Output the rice side dish name and price.
     */
    printf("    Rice:                %5.2f\n", side_dish_price);

} /* if (side_dish_item_code == rice_item_code) */
else if (side_dish_item_code == onion_rings_item_code) {
    /*
     * Output the onion rings side dish name and price.
     */
    printf("    Onion Rings:          %5.2f\n", side_dish_price);
} /* if (side_dish_item_code == onion_rings_item_code) */
```

WE URGE YOU TO USE THE CODE JUST ABOVE IN YOUR PP#4!

Notice that, in addition to the properties of just an `if` clause, above:

- Again, the comment preceding the entire `if` block describes the purpose of the `if` block as a whole, not just the purpose of one of the `if` block's individual clauses.
- There is no comment immediately preceding any of the `else if` statements.
- The condition in the comment immediately following and on the same line as each `else if` clause's block close is the condition of the associated `else if` statement.
- The comment following, and on the same line as, each clause's block close **DOESN'T** include the keyword `else`.
- Again, the comment on the same line as the block close is **ABSOLUTELY FORBIDDEN** to continue on to the next line.
- Again, in the event that the `if` condition or an `else if` condition is very long, then in the comment on the associated block close line of that clause, you can use the first part of the condition, followed by an ellipsis (three periods).
- Again, the comments for the statements inside the `else if` clause are indented farther, so that the asterisk at the beginning of each line of the comment lines up with the statement that the comment describes.
- Again, there is no comment immediately preceding each block close, because block closes are not statements.

(4) In the event that the `if` block has an `if` clause, one or more `else if` clauses, as well as an `else` clause:

```
/*
 * Output the drink on the bill.
 */
if (drink_item_code == coffee_code) {
    /*
     * Output coffee as the drink.
     */
    printf("    Coffee:                $%5.2f\n",
           drink_price);
} /* if (drink_item_code == coffee_code) */
else if (drink_item_code == soda_code) {
    /*
     * Output soda as the drink.
     */
    printf("    Soda:                $%5.2f\n",
           drink_price);
} /* if (drink_item_code == soda_code) */
else {
    /*
     * Output tea as the drink.
     */
    printf("    Tea:                $%5.2f\n",
           drink_price);
} /* if (drink_item_code == ice_tea_code)...else */
```

DON'T USE THE CODE JUST ABOVE IN YOUR PP#4!

Notice that, in addition to the properties of all of the previous cases:

- The comment preceding the entire `if` block describes the purpose of the `if` block as a whole, not just the purpose of the `if` clause.
- The condition in the comment immediately following and on the same line as the `else` clause's block close is the condition of the **FINAL** `else if` clause, **followed by an ellipsis and the keyword `else`.**