Conditional branching (if-then-else statements)

It's time to break away from the boring stuff. Programs become much interesting and useful when they can make decisions.

By posing and answering a simple question: Is something true? the program can branch into two possible path, depending on whether the answer is true or false.

The basic program structure:

```
If ( some conditional expression ) {
    Take some action if the expression is true;
}
else {
    Take a different action if the expression is false;
}
```

The else part is optional, as we will see in some of the examples.

The conditional statement

The conditional statement must be evaluated to an answer of either “true” or “false”. The program will actually be looking to see of the conditional statement evaluates to “0” (false) or “not 0” (i.e. true, which includes every other possible outcome). The statement will *usually* compare two values — the comparison will evaluate to either true or false. If the result is true, then the following set of instructions is implemented. And the result is false, the program jumps to the instructions following the “else” part and executes those. (If there is an else part.)

`a < b`	a is less-than b	`if( theNum < 12 )`
`a > b`	a is greater-than b	`if( bigBoy > 100.0 )`
`a <= b`	a is less-than or equal to b	`if( aBlob <= theSplat )`
`a >= b`	a is greater-than or equal to b	`if( myCash => 42.42 )`
`a == b`	a is equal to b (exactly)	`if( theTime == HIGH_NOON )`
`a != b`	a is not equal to b	`if( myPickle != 17 )`

Be careful!

`a <= b` not `a =< b`

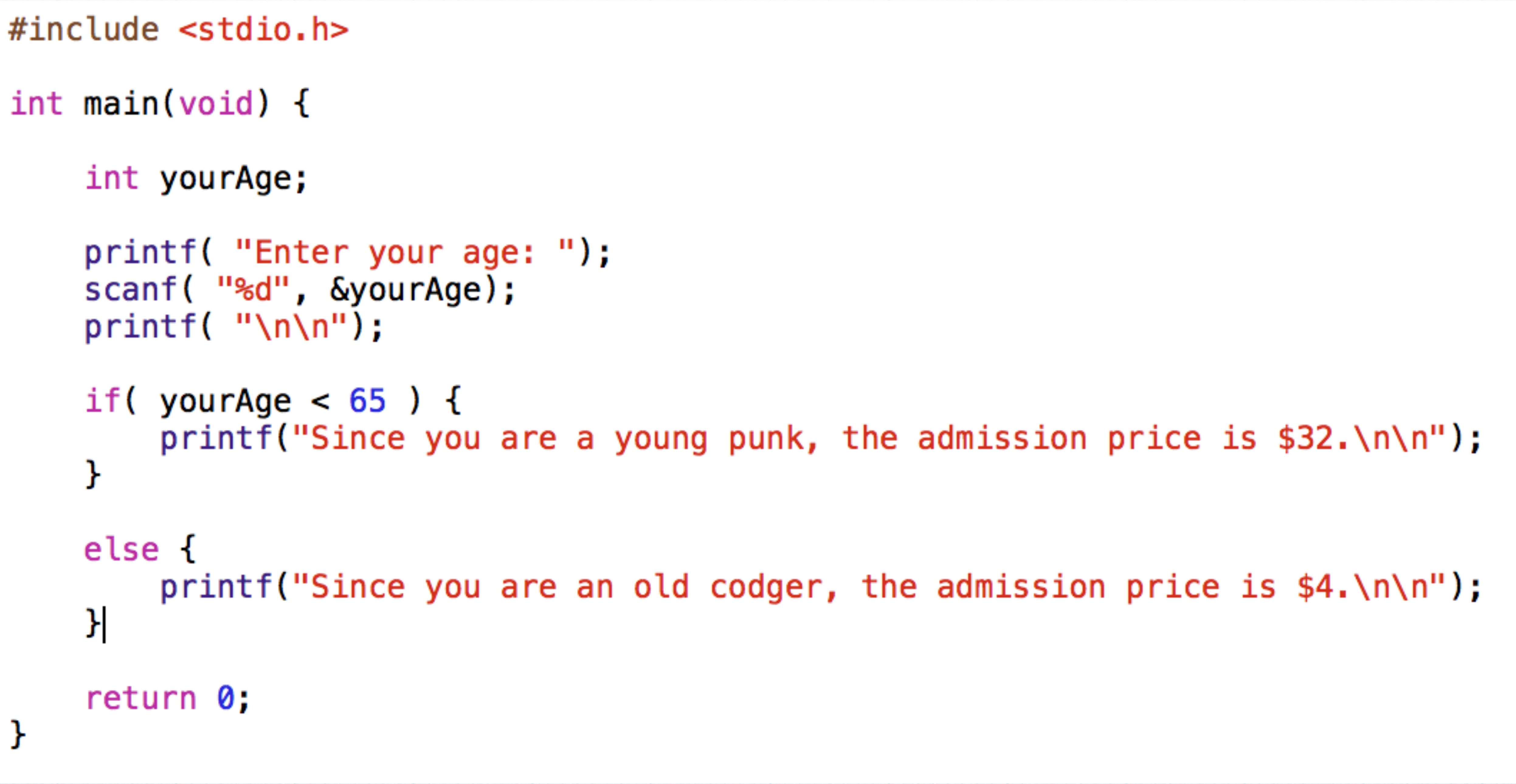
`a >= b` not `a => b`

`a == b` not `a = b` Major source of errors!!

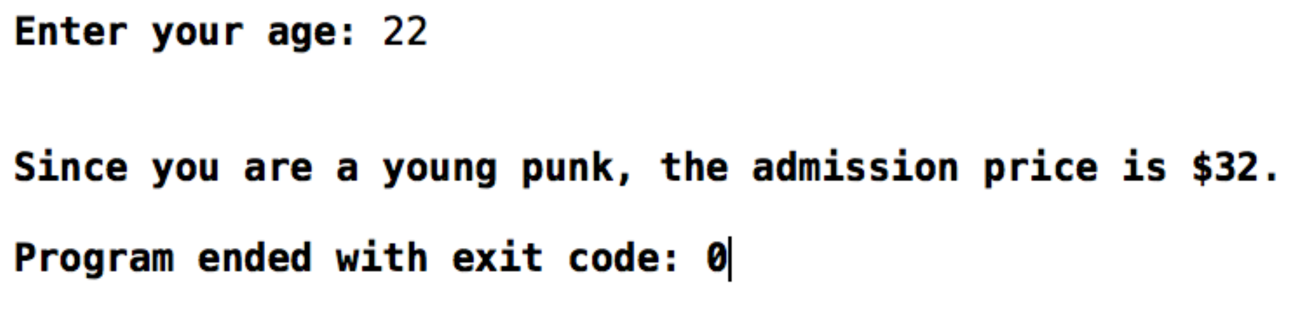
`a == b` Works well with integers, but tricky with float or double.

The optional code that follows the “if” portion is lumped together inside of braces. These are just regular commands and anything can go in here. (Including more “if-then-else” statements.)

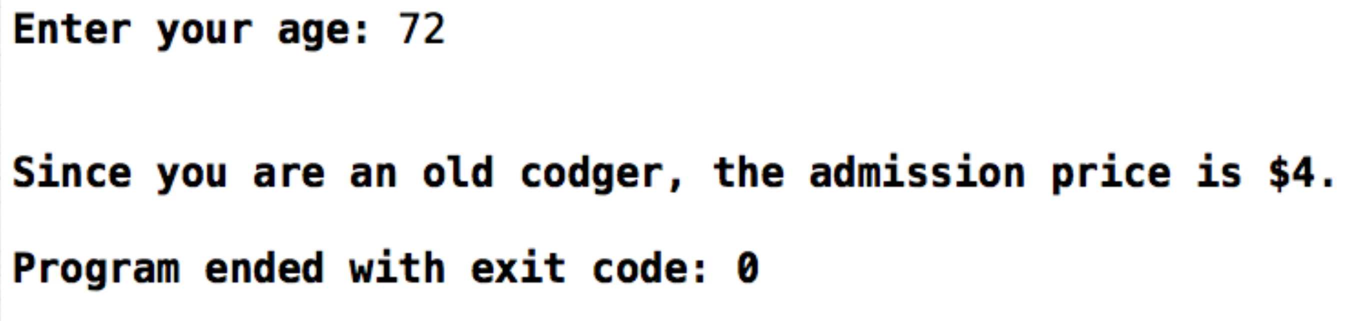
Simple example



```c
#include <stdio.h>

int main(void) {

    int yourAge;

    printf( "Enter your age: ");
    scanf( "%d", &yourAge);
    printf( "\n\n");

    if( yourAge < 65 ) {
        printf("Since you are a young punk, the admission price is $32.\n\n");
    }

    else {
        printf("Since you are an old codger, the admission price is $4.\n\n");
    }

    return 0;
}
```



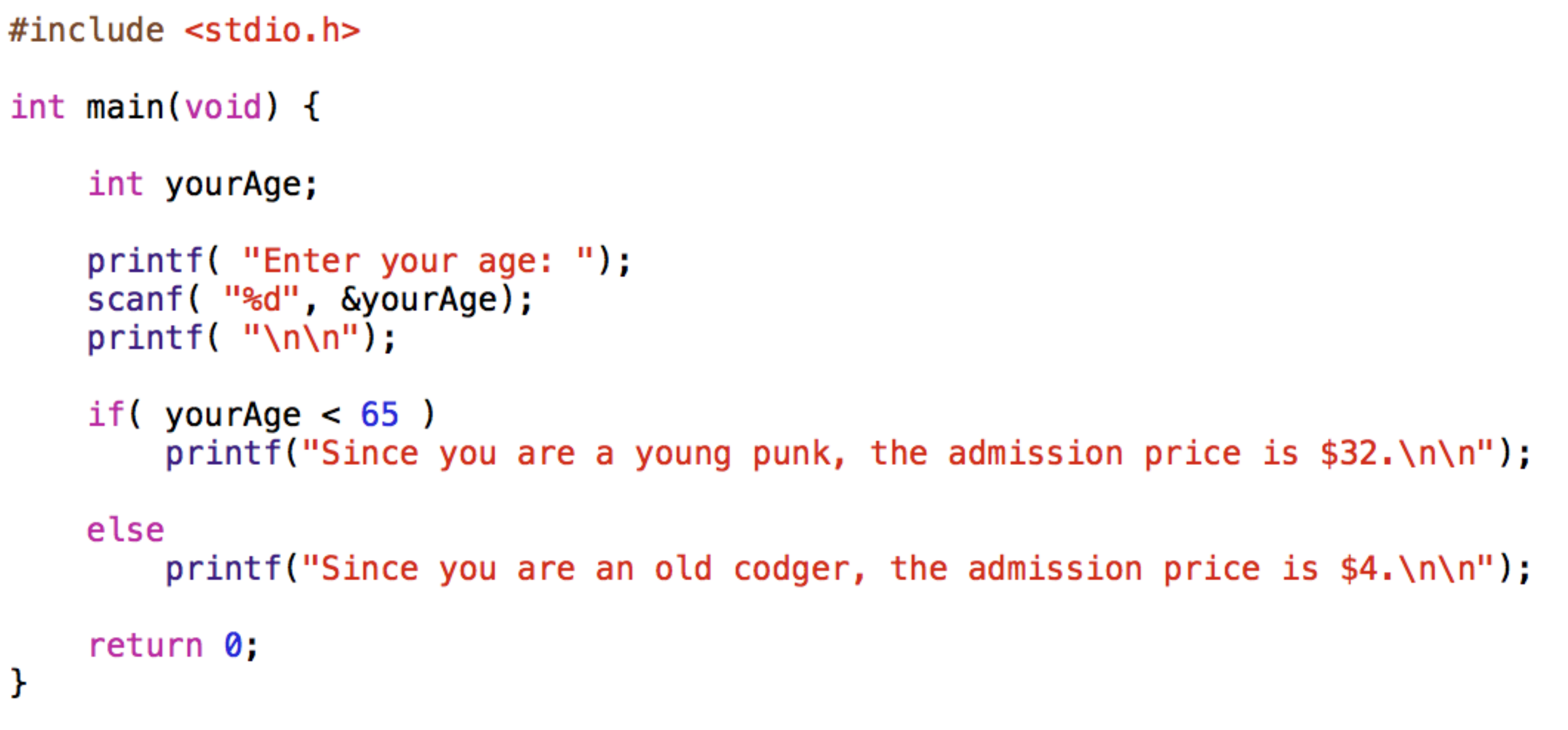
```
Enter your age: 22


Since you are a young punk, the admission price is $32.

Program ended with exit code: 0
```



```
Enter your age: 72


Since you are an old codger, the admission price is $4.

Program ended with exit code: 0
```

Style consideration

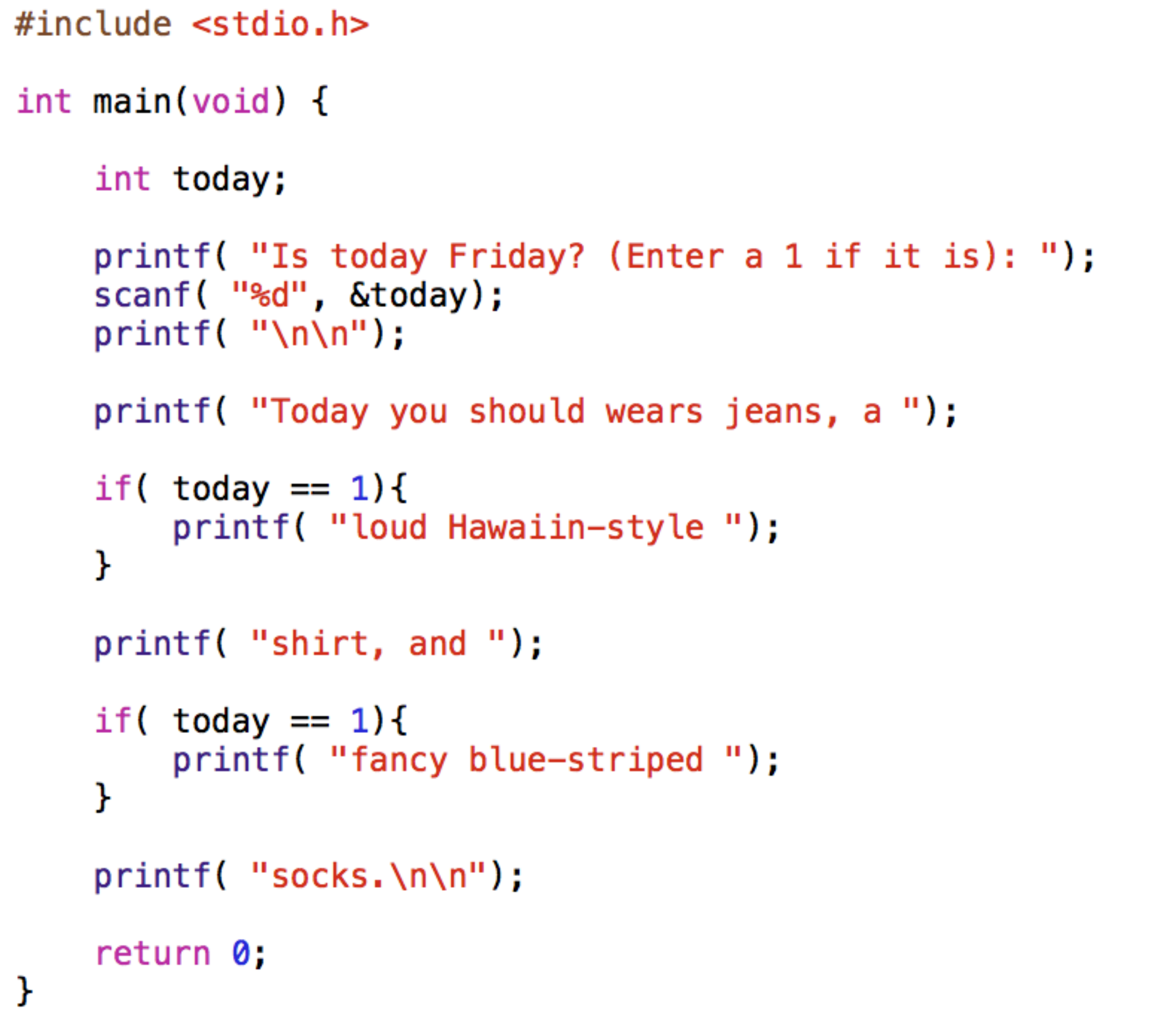
If there is only one statement following either the “if” or the “else” part of the branching command, the braces are optional.



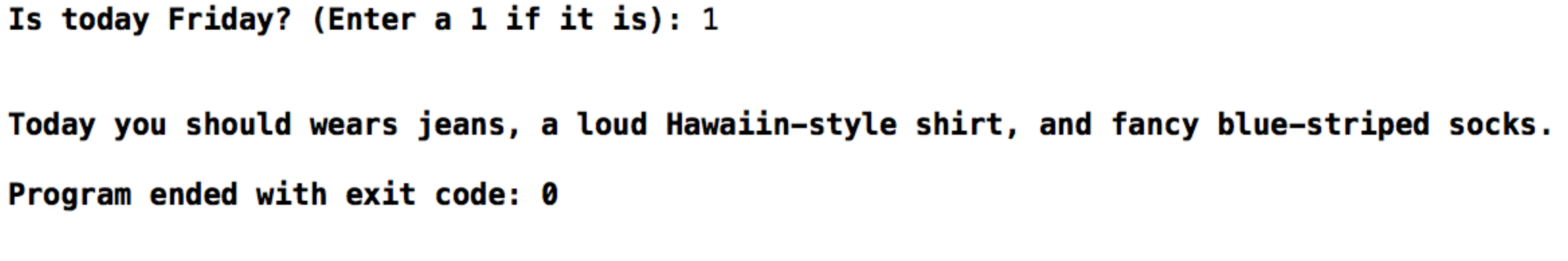
```
#include <stdio.h>

int main(void) {

    int yourAge;

    printf( "Enter your age: ");
    scanf( "%d", &yourAge);
    printf( "\n\n");

    if( yourAge < 65 )
        printf("Since you are a young punk, the admission price is $32.\n\n");

    else
        printf("Since you are an old codger, the admission price is $4.\n\n");

    return 0;
}
```

This will compile and run just fine. But be careful, it is easy to make mistakes. If you add a command in either spot, without adding the braces to group things, your program will not run correctly. Zybooks recommends *always* using braces. K&R (and GT) goes without braces for single line branches. It’s your call.

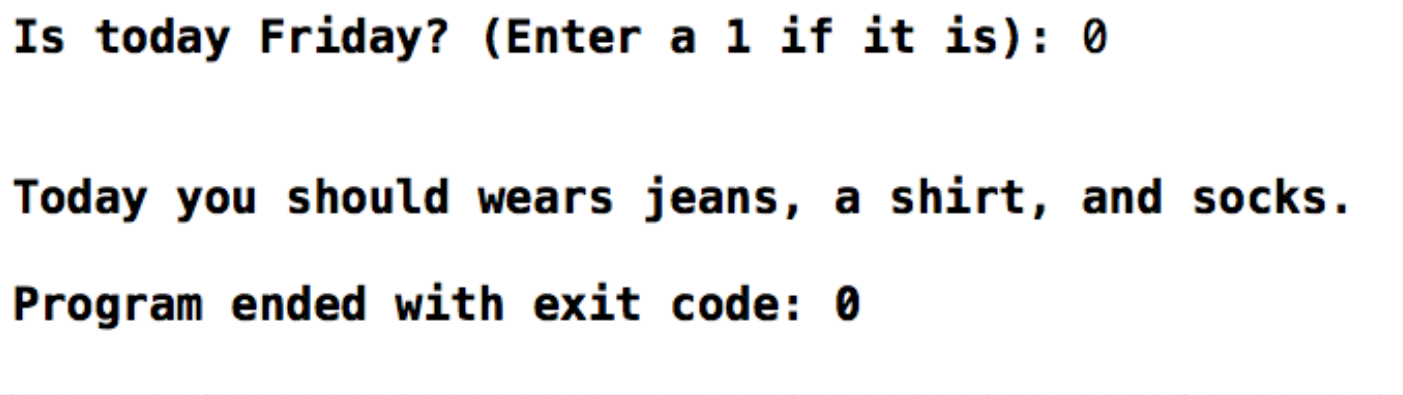
Simple example - “else” is optional



```
#include <stdio.h>

int main(void) {

    int today;

    printf( "Is today Friday? (Enter a 1 if it is): ");
    scanf( "%d", &today);
    printf( "\n\n");

    printf( "Today you should wears jeans, a ");

    if( today == 1){
        printf( "loud Hawaiin-style ");
    }

    printf( "shirt, and ");

    if( today == 1){
        printf( "fancy blue-striped ");
    }

    printf( "socks.\n\n");

    return 0;
}
```

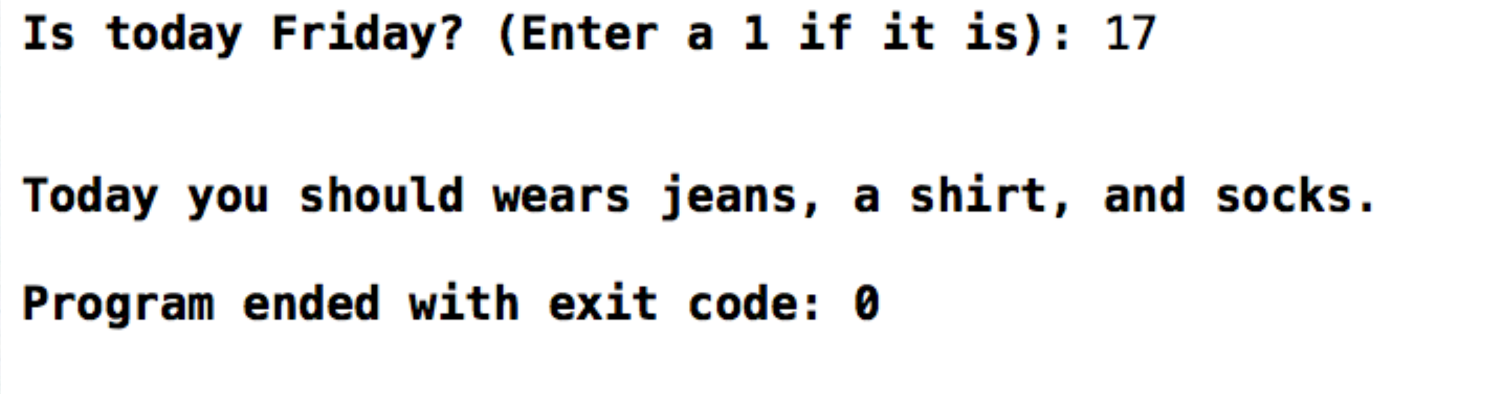


```
Is today Friday? (Enter a 1 if it is): 1


Today you should wears jeans, a loud Hawaiin-style shirt, and fancy blue-striped socks.

Program ended with exit code: 0
```



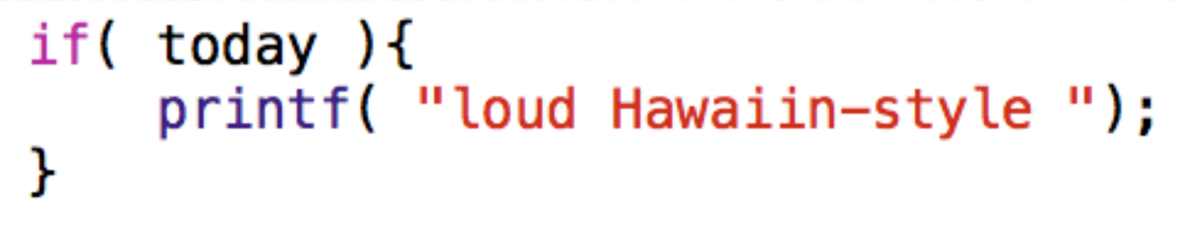
```
Is today Friday? (Enter a 1 if it is): 0


Today you should wears jeans, a shirt, and socks.

Program ended with exit code: 0
```

Note that entering anything other than “1” will result in the conditional being evaluated as false.



```
Is today Friday? (Enter a 1 if it is): 17


Today you should wears jeans, a shirt, and socks.

Program ended with exit code: 0
```

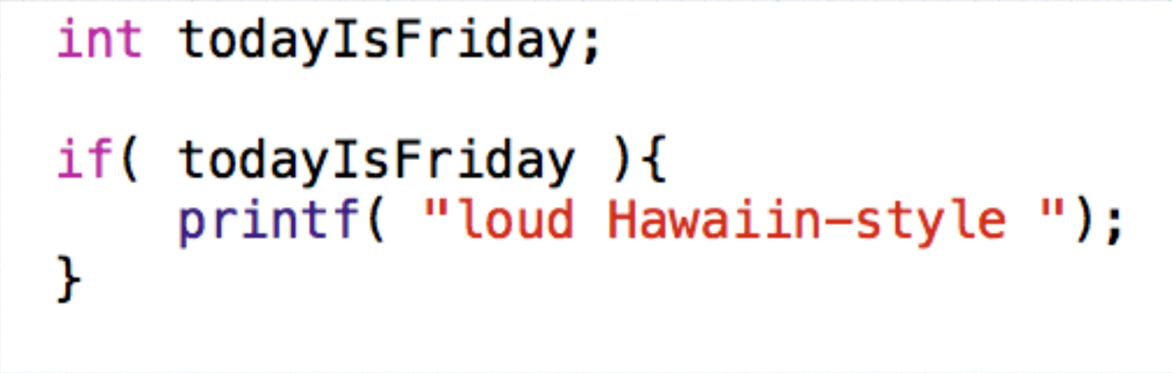
The conditional does not have to be a strict comparison between two quantities. It can a simple variable that can have a value of “1” (true) or something else (false).



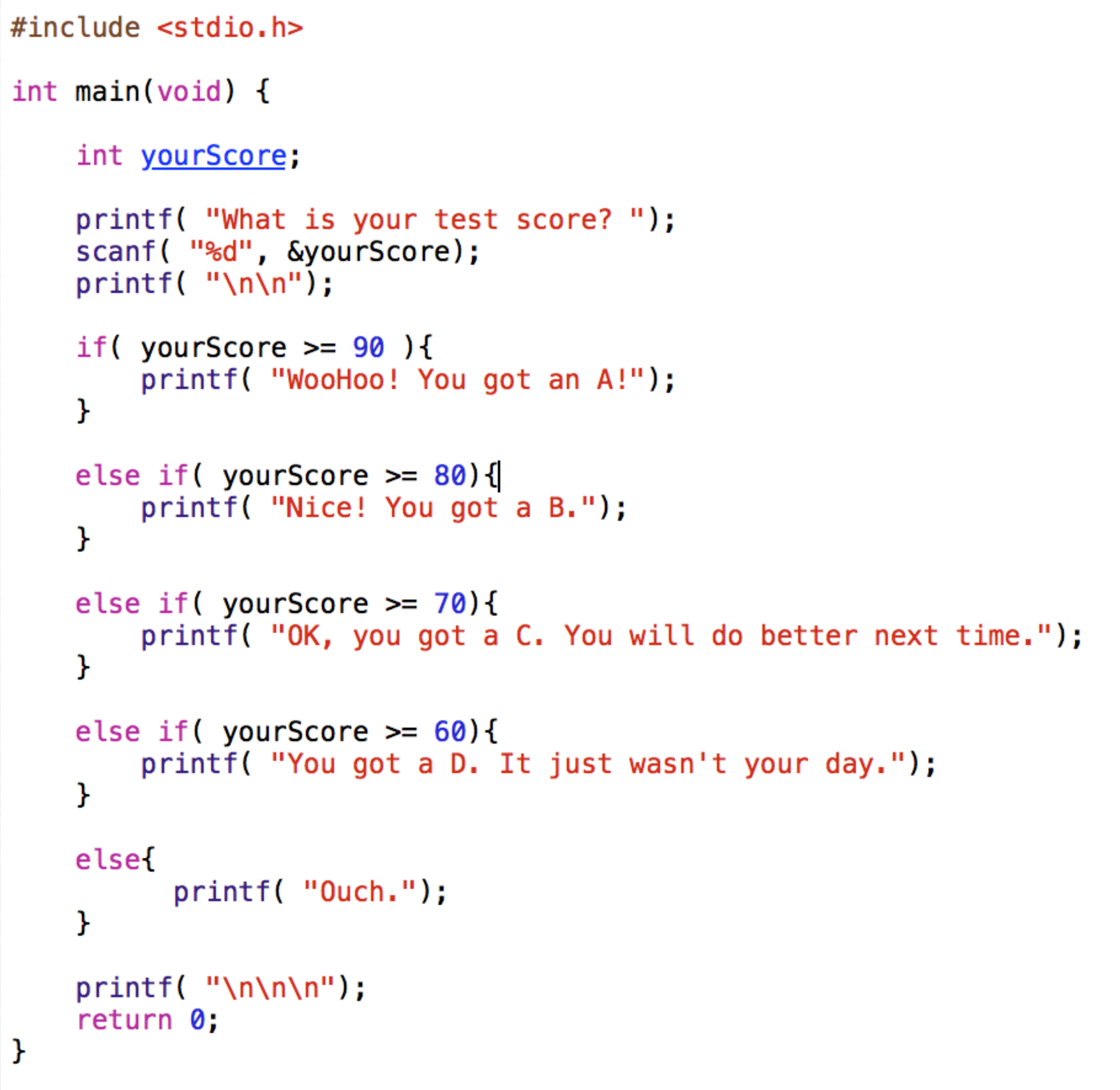
```
if( today ){
    printf( "loud Hawaiin-style ");
}
```

When you are using a variable to provide simple true/false information, it is sometimes known a Boolean data type.

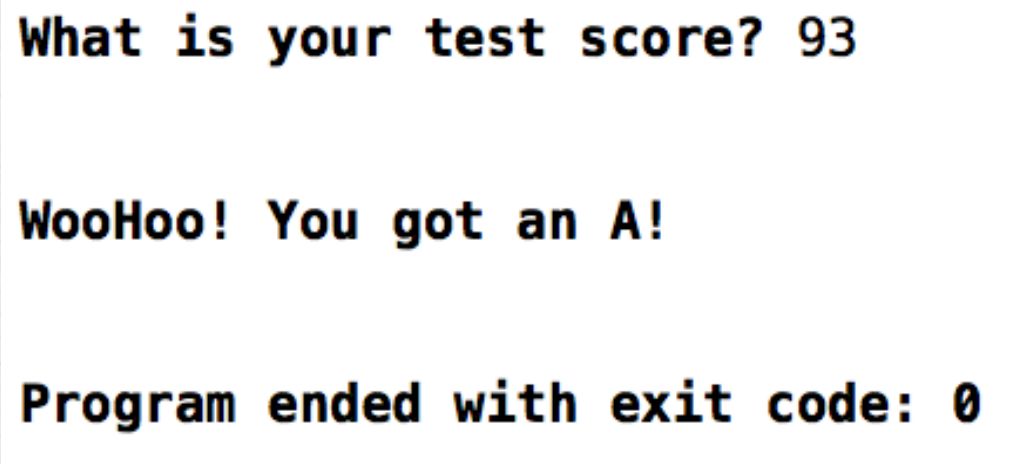
I find this handy in many situations, but reading the program can be confusing. If you are going a variable in this Boolean fashion, you might consider changing the name.



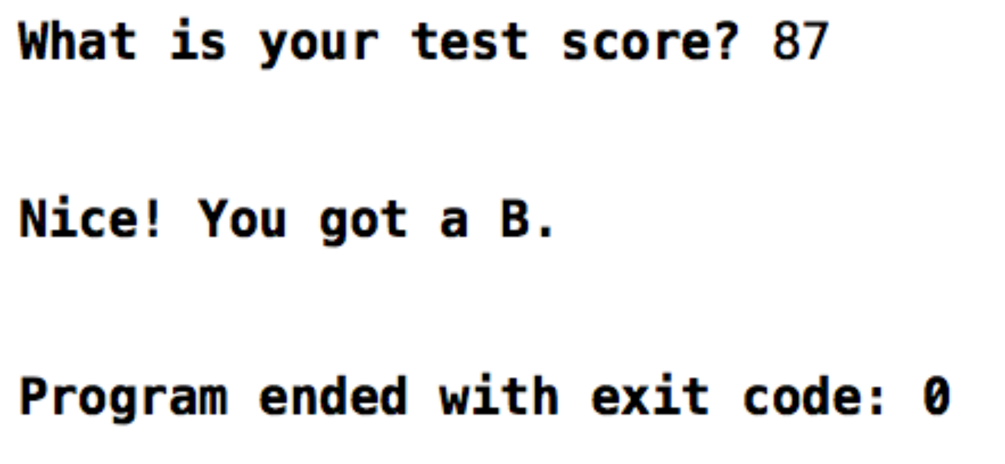
```
int todayIsFriday;

if( todayIsFriday ){
    printf( "loud Hawaiin-style ");
}
```



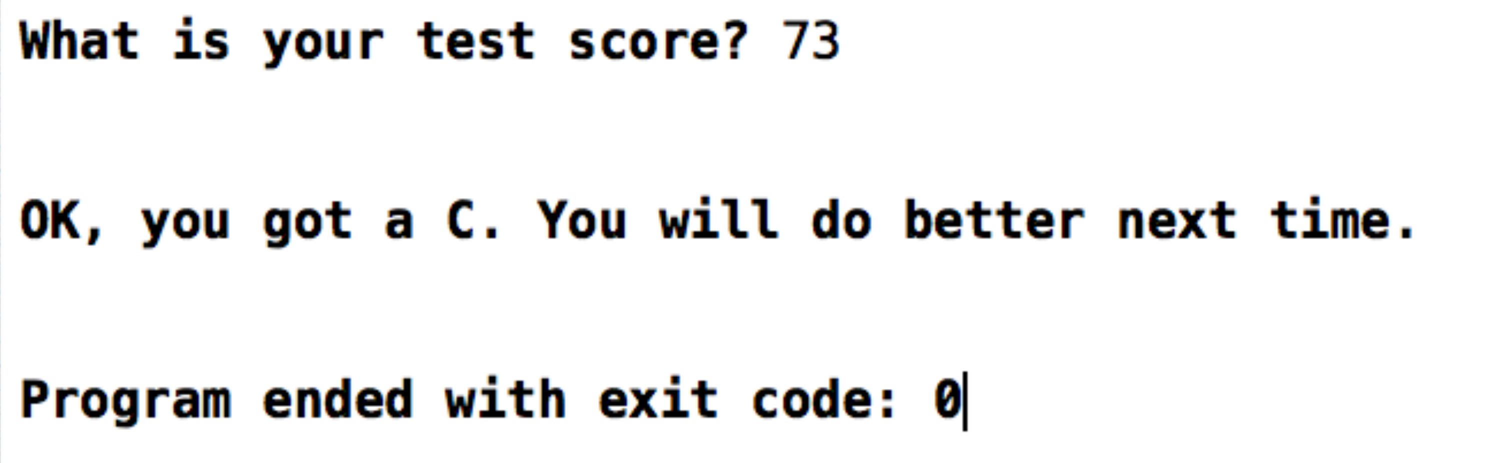
```c
#include <stdio.h>

int main(void) {

    int yourScore;

    printf( "What is your test score? ");
    scanf( "%d", &yourScore);
    printf( "\n\n");

    if( yourScore >= 90 ){
        printf( "WooHoo! You got an A!");
    }

    else if( yourScore >= 80){
        printf( "Nice! You got a B.");
    }

    else if( yourScore >= 70){
        printf( "OK, you got a C. You will do better next time.");
    }

    else if( yourScore >= 60){
        printf( "You got a D. It just wasn't your day.");
    }

    else{
          printf( "Ouch.");
    }

    printf( "\n\n\n");
    return 0;
}
```



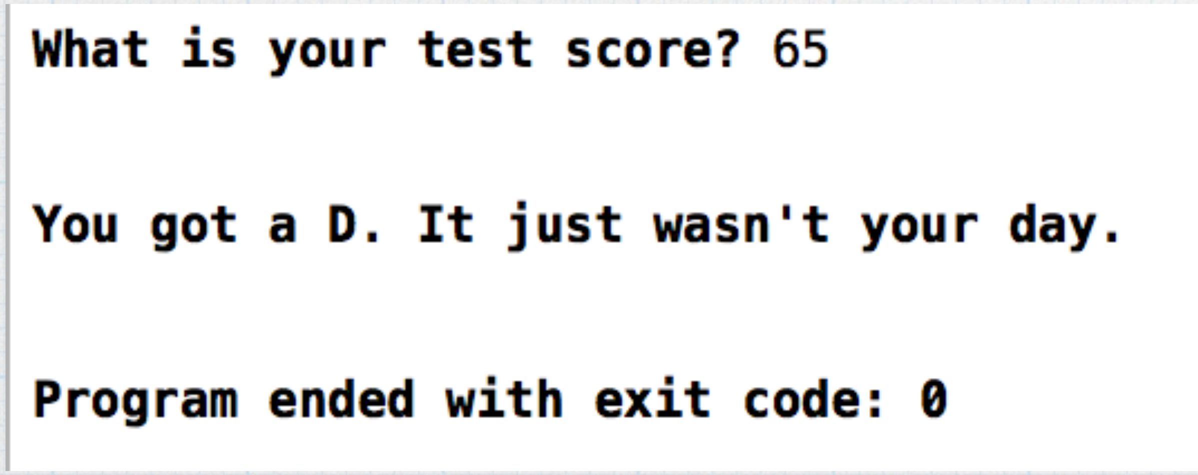
```
What is your test score? 93

WooHoo! You got an A!

Program ended with exit code: 0
```



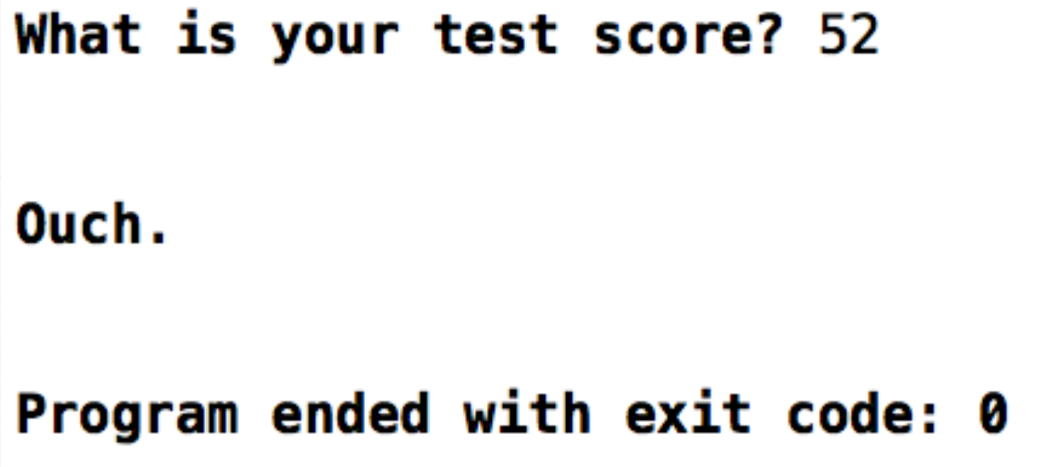
```
What is your test score? 87

Nice! You got a B.

Program ended with exit code: 0
```



```
What is your test score? 73

OK, you got a C. You will do better next time.

Program ended with exit code: 0
```



```
What is your test score? 65

You got a D. It just wasn't your day.

Program ended with exit code: 0
```



```
What is your test score? 52

Ouch.

Program ended with exit code: 0
```