

# Spring 2022 courses

## IN PROGRESS cc quiz 5

### Settings

- A question has only ONE right answer unless otherwise noted
- The quiz covers the recent content (weeks 6+7)
- After the first play, the quiz will be opened for unlimited play
- Let me know if you have any comments or corrections

### Be the compiler!

The following program statements should print out 300 but it does not even compile! What's wrong?

```
int fvoid1=100, fvoid2=200, void;  
printf("%dn", void = fvoid1 + fvoid2);
```

*This question has more than one correct answer.*

TRUE:

- The keyword void is protected, cannot be a variable
- There is a \ missing in the formatter %dn

FALSE:

- The variables in printf are not recognized
- The formatter should be %f

Feedback: the correct statements would be

```
int fvoid1=100, fvoid2=200, void0;  
printf("%d\n", void0 = fvoid1 + fvoid2);
```

300

### Be the compiler!

The function call hello() below should print out hello world but it does not though the program compiles and runs. The warning is "hello() defined but not used"

```
#include <stdio.h>  
  
int main(void) {  
    int hello(void) {  
        puts("hello world");  
        return 0;  
    }  
}
```

```
    }  
    return 0;  
}  
// function call  
hello();
```

TRUE:

- The function `hello()` needs to be called inside `main`.

FALSE:

- There are too many `return 0;` commands - only one is needed.
- This is an Emacs problem. It works on the command line.
- The function `puts()` does not print. It should be `printf()`

## Meta matters

In Org-mode, `#+TITLE:` must start in the first column of the file to be recognized.

TRUE

## Be the compiler!

The following program compiles and runs but it does not print the right results - it always prints 1, no matter what is entered. What is the problem?

```
int val, input;  
val = scanf("%d", &input);  
printf("%d\n", val);
```

TRUE:

- `input` should be printed, not `val`

FALSE:

- The variable `val` needs to be a floating point number
- The `printf()` formatter needs to be `%f`
- The variable `input` is actually a constant

Feedback: the correct program is:

```
int input;  
scanf("%d", &input);  
printf("%d\n", input);
```

If you look at the `scanf` manual page, you read that it "returns the number of input items successfully matched and assigned." For `val=scanf("%d",&input);` only 1 int was scanned, not the integer value of the scanned character (`input`).

## Help me escape!

Match the `printf()` statement and the output.

---

<code>printf("hi there");</code>	hi there
<code>printf("\\"hi there\\");</code>	"hi there"
<code>printf("This is a slash: \\");</code>	This is a slash:
<code>printf("This is a slash: \\\");</code>	This is a slash: \

---

## Reading input

Suppose that we call `scanf` as follows:

```
scanf("%d%f%d", &i, &x, &j);
```

If the user enters

```
10.3 5 6
```

What will be the values of `i`, `x`, and `j` after the call?

Assume that `i` and `j` are uninitialized `int` variables, and `x` is an uninitialized `float` variable.

TRUE:

- `i=10`, `x=.300000`, `j=5`

FALSE:

- `i=10`, `x=.300000`, `j=5`
- `i=10`, `x=5.000000`, `j=6`
- `i=10.3`, `x=0`, `j=6`

Feedback: program to check

```
int i, j;
float x;
scanf("%d%f%d", &i, &x, &j);
printf("i = %d, x = %f, j = %d\n", i, x, j);
```

```
i = 10563276, x = 0.000000, j = 67
```

The input is parsed as 10 .3 5 because that is the input that was formatted in `scanf()`. This program would print out what we entered:

```
int x,j;
float i;
scanf("%f%d%d", &i, &x, &j);
printf("i = %.1f, x = %d, j = %d\n", i, x, j);
```

```
i = 10.3, x = 5, j = 6
```

## Reading input

Suppose that we call `scanf` as follows:

```
scanf("%f%d%f", &x, &i, &y);
```

If the user enters

```
12.3 45.6 789
```

What will be the values of `x`, `i`, and `y` after the call?

Assume that `x` and `y` are uninitialized `float` variables, and `i` is an uninitialized `int` variable.

TRUE:

- `x=12.300000, i=45, y=.600000`

FALSE:

- `x=12.300000, i=45, y=789.000000`
- `x=12.3, i=45.789, y=0.000000`
- `x=12.30, i=45, y=789.00`

Feedback: program to check

```
int i;  
float x,y;  
scanf("%f%d%f", &x, &i, &y);  
printf("x = %f, i = %d, y = %f\n", x, i, y);
```

```
x = 12.300000, i = 45, y = 0.600000
```

The input is parsed as 12.3 45 .6 because that is the input that was formatted in `scanf()`. This program would print out what we want to see.

```
int y;  
float x,i;  
scanf("%f%f%d", &x, &i, &y);  
printf("i = %f, x = %f, j = %d\n", x, i, y);
```

```
i = 12.300000, x = 45.599998, j = 789
```

**These statements are illegal (program will not compile)**

Tip: according to the C standard, the compiler must replace each comment with a single space character.

```
int a,b;  
a/**/b = 0;
```

TRUE

Feedback: when removing the comments and replacing them with single spaces, the statement results in `ab=0`, which is illegal. The compiler expects a semi-colon `;` before the statement `b=0`;

**When a program reaches the end of the main function without executing a return statement, it will not terminate.**

FALSE

Feedback: The return statement is not mandatory. If it is missing, the program will still terminate but it may generate a warning. If `main` is declared to return an `int`, the program returns `0` to the operating system; otherwise, the program returns an unspecified value.

**Be the compiler!**

The hello world program below works well but the compiler may return a warning, especially when run with the `-Wall` option:

```
#include <stdio.h>  
  
int main(void)  
{  
    printf("hello world\n");  
}
```

What is needed to make the warning go away?

TRUE:

- Add `return 0;` right at the end before `}`.

FALSE:

- Declare "hello world" as a character constant
- Use `puts()` instead of `printf()`
- Nothing. Compilers are famously picky for no reason.

Feedback: The return statement is not mandatory. If it is missing, the program will still terminate but it may generate a warning. If `main` is declared to return an `int`, the program returns `0` to the operating system; otherwise, the program returns an unspecified value.

Author: Marcus Birkenkrahe

Created: 2022-02-27 Sun 22:07

[Validate](#)