

Types, Arithmetic Operators and precedence

1. Assume that the following variable declarations have been made:

```
int a = 4;  
int b = 8;  
double c = 2.5;
```

What is the value of each of the following expressions? Recall operator precedence. Here is an online precedence table for your reference (ignore the operators not covered yet):

https://www.tutorialspoint.com/cprogramming/c_operators_precedence.htm

- a) $2 + 3 * a = 14$
- b) $a * c - 2 = 8.0$
- c) $a / b * c = 0.0$
- d) $a * c / b = 1.25$

2. Assume that the following variable declarations have been made:

```
int f;  
int g = 3;
```

Assume that the following statements are executed one after another in the given sequence. What is the value of the variable `f` after each statement has executed? We have provided a trace table for you to fill in with the current values of `f` and `g` filled in for you.

```
f = f;  
f = g * 2;  
f = g * 1.5;  
f = f + 2;
```

f	g
-	3
-	
6	
4	
6	

3. Suppose that you've been asked to write a program that creates a variable with an initial weight value in kilograms and prints the corresponding weight in pounds on the screen. We assume a conversion rate of 2.2 pounds per kilogram. Let's assume that a friend of yours has written the following program:

```
/*
 * Author: Ein Freund
 * Purpose: takes a weight in kilograms and converts it to pounds.
 */

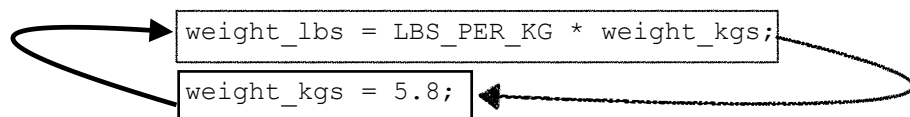
#include <stdio.h>

#define LBS_PER_KG 2.2

int main( void ) {
    double weight_kgs;
    double weight_lbs;

    weight_lbs = LBS_PER_KG * weight_kgs;
    weight_kgs = 5.8;
    printf( "Corresponding weight in lbs: %f\n", weight_lbs);

    return 0;
}
```



The program compiles and runs but, unfortunately, it doesn't produce the correct output. Identify the problem. Then figure out how you would explain to your friend why the program doesn't work and what needs to be done to fix it.

The statement that assigns a value to `weight_lbs` is in the wrong place. When this statement is in its current position, `weight_kgs` has a garbage value assigned to it and so a garbage value is assigned to `weight_lbs`. We need to move this statement so that it immediately follows the statement that assigns the value of 5.8 to `weight_kgs` and that value will be used to compute `weight_lbs`.

4. Assume that the following variable declarations have been made:

```
int a = 4;
int b = 8;
```

What is the value of each of the following expressions?

- a) `(double) a / b` **evaluates to 0.5**
- b) `(double) (a / b)` **evaluates to 0.0**

5. Suppose you've been asked to write a program that prompts the user for a Celsius temperature and converts it to Fahrenheit. Let's assume that a friend of yours has written the following program:

```
/*
 * Author: Pat Gear
 * Purpose: defines a temperature variable in Celcius and
 * prints the corresponding temperature on the Fahrenheit scale
 */
#include <stdio.h>

int main( void ) {
    double celsius = 0;
    double fahrenheit;

    fahrenheit = 9 / 5 * celsius + 32;
    printf( "Corresponding Fahrenheit temperature: %f\n", fahrenheit );

    return 0;
}
```

Your friend tested the program by assigning a value of 0 for Celsius and got the expected value of 32 Fahrenheit printed on the screen – yay! BUT... the program is not correct.

- What's wrong with the program and what would you do to fix it?
9/5 is Integer division. Fix it by forcing either or both the 9 and 5 to floating point number:
fahrenheit = 9.0 / 5.0 * celsius + 32;
- Why did the program pass the above-mentioned test?
Any number multiplied by 0 is 0, therefore 9.0/5.0 could be replaced with any value and the formula would still compute the Fahrenheit for 0 degrees Celsius correctly
- What other values of Celsius should be tried to verify that the program will work correctly for any value of celsius?
Check a non-zero value for Celsius. Ie Celsius of 100 is 212 Fahrenheit

6. Assume that the following variable declarations have been made and that the statements below are executed in the order shown. What is the value of `f` and `g` after each statement? Use a dash (—) for garbage values. A trace table has been provided for you.

```
int f;

int g = 3;

f = 1 * 2.1;

f *= g;

f++;

--f;

f = g--;

f = --g;
```

f	g
—	
	3
2	
6	
7	
6	
3	2
1	1

7. What is the output of the following program?

```
#include <stdio.h>

void foo();
void bar();

int main( void ) {
    bar();
    return 0;
}

void foo( ) {
    int x = 10;
    printf( "foo 1, x: %d\n", x);
    x += 100;
    printf( "foo 2, x: %d\n", x);
}

void bar( ) {
    int x = 20;
    printf( "bar 1, x: %d\n", x);
    x++;
    foo();
    printf( "bar 2, x: %d\n", x);
}
```

bar 1, x: 20

foo 1, x: 10

foo 2, x: 110

bar 2, x: 21

8. Design a function named `celebrate` that will print the following. Declare a variable within the function called `age` and initialize it to your age. Be sure to use the variable in your print statement.

```
\*****/  
"Happy Birthday! You are 19!"  
/*****\
```

Don't forget to call the function from `main` so that it will execute when you run your program!

```
/*  
 * Author: CSC 111 Instructor  
 * Date: Jan 1, 2020  
 * Purpose: prints a birthday message  
 */  
#include <stdio.h>  
  
void celebrate();  
  
int main( void ) {  
    celebrate();  
  
    return 0;  
}  
  
/*  
 * Purpose: prints Happy Birthday to someone age 19  
 */  
void celebrate() {  
    int age = 19;  
    printf("\n\*****/\n");  
    printf( "\nHappy Birthday! You are %d!\n\n", age);  
    printf("/*****\\n");  
}
```