

1 Dimensional weight reconsidered

1. [] Checklist before you start coding:
 - Go to a working directory where you are allowed to save files
 - If you are in AppData/Roaming, change to your home directory
 - Open a terminal and navigate to your working directory
2. [] Create an Org-mode file "dweight.org"
3. [] Add one named C code block (use <s [TAB] to do this)
4. [] Name the code block dweight
5. [] Start with the code that we created last week, or take the code from [dweight.c from GitHub](#) (in [cc100/3_basics/src](#)).
6. []

Check: your code block should now look like this:

```
#include <stdio.h>

int main(void)
{
    const int INCHES_PER_POUND = 166;
    int height, length, width, volume, weight;

    height = 8;
    length = 12;
    width = 10;
    volume = height * length * width;
    weight = (volume + INCHES_PER_POUND-1) / INCHES_PER_POUND;

    printf("Dimensions: %dx%dx%d\n", length, width, height);
    printf("Volume (cubic inches): %d\n", volume);
    printf("Dimensional weight (pounds): %d\n", weight);

    return 0;
}
```

7. [] You do not need any code block header arguments except c and :results output.
8. [] Note that the number 166 is now defined as const.
9. []

Run the code block in Emacs with C-c C-c to make sure it is correct. You should get this output:

```
#+RESULTS: dweight
: Dimensions: 12x10x8
: Volume (cubic inches): 960
: Dimensional weight (pounds): 6
```

10. [] Copy the code block into a second code block, and name it dweight1.
11. [] Alter the code so that the three variables height, length, and width have to be given as input:
 - Add three printf statements asking for these variables
 - After each of these, add a scanf statement to take the input
 - You must delete the assignments to these variables
 - Make sure that you use the correct format identifier %d
12. []

Your code block should now look like this (without the comments):

```
#include <stdio.h>

int main(void)
{
    // declare variables and constants
    const int INCHES_PER_POUND = 166;
    int height, length, width, volume, weight;

    // Enter input
    printf("Enter the box height: ");
    scanf("%d", &height);
    printf("Enter the box length: ");
    scanf("%d", &length);
    printf("Enter the box width: ");
    scanf("%d", &width);

    // compute volume and dimensional weight
    volume = height * length * width;
    weight = (volume + INCHES_PER_POUND-1) / INCHES_PER_POUND;

    printf("Dimensions: %dx%dx%d\n", length, width, height);
    printf("Volume (cubic inches): %d\n", volume);
    printf("Dimensional weight (pounds): %d\n", weight);

    return 0;
}
```

13. []

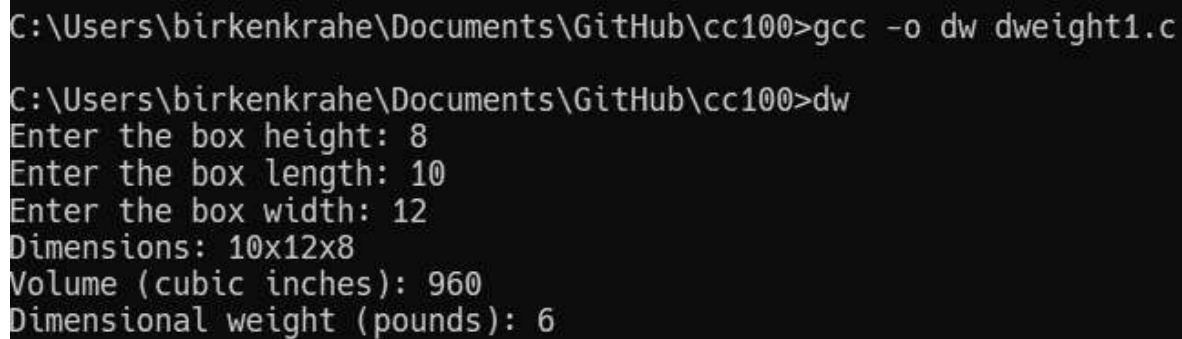
Add the necessary tangle information to the code block header:

```
:tangle dweight1.c
```

14. [] Tangle the code block with the key sequence C-c C-v t to get the C source code file dweight1.c

15. []

Compile and run it on the Windows command line, and you're done!



```
C:\Users\birkenkrahe\Documents\GitHub\cc100>gcc -o dw dweight1.c

C:\Users\birkenkrahe\Documents\GitHub\cc100>dw
Enter the box height: 8
Enter the box length: 10
Enter the box width: 12
Dimensions: 10x12x8
Volume (cubic inches): 960
Dimensional weight (pounds): 6
```

Figure 1: sample result on the Windows CMD line

16. [] Alternatively, you can pass the input data to the file using the redirection operator < - then you can run the file inside Emacs (either on the shell, or using the code block header). Examples in GitHub:

cc100/practice/[cmdline.org](https://github.com/cc100/practice).

17. []

You can submit the Org-file to me **via Email** for extra credit - with the three lines at the header to identify your work:

```
#+TITLE: Dimensional weight  
#+AUTHOR: your name  
#+HONOR: pledged
```

Created: 2022-02-28 Mon 09:15

Validate