

Clayton Price

 Search this site[Courses](#) [Missouri S&T](#) [CS Department](#) [Help](#) [Guides](#)[Courses](#) > [CS 1570 \(formerly 053\) - Intro to Programming C++](#) > [Homework Assignments](#) > [2017](#) > [Spring](#) >

Assignment 02

Due: Friday, February 3, 2016 at noon 100 pts

For this assignment, you will submit a single C++ compilable file containing a program written in C++. Remember, to electronically submit a file to for this course, first change to the directory in which the file resides and then type in, at the UNIX prompt, the command:

```
cssubmit 1570 section_letter assignment_number
```

Be sure that only the file you want to submit is in that directory - make a directory for every assignment! Remember that the submit system will deliver every .cpp file in the current directory to me, so you should only have the file for hw 2 in that directory. Name your file a meaningful name and **give it a .cpp extension** since you will be compiling it. Also, make sure that you compile and run your program using the GNU (g++) compiler before submitting to make sure that it will work for the submit script.

Foreground: This semester you will be writing code in C++ to help solve real-life problems, advance the cause of humanity, destroy desperation, solve world hunger, and generally make the planet a better place to live. Mostly, your code is going to be used by this distinguished gentleman, Milhouse Van Houten. Now, he may look like a dork, but the reality is that well, ok, he is a dork. But Milhouse will keep you busy with his very much real-life problems.

Background: Chief among Millhouse's problems is his allergies. He's allergic to several things and the reactions can be a real problem. Has to take lots of pills to deal with it. But the real difficulty here is that he has to "calculate" the number of pills to take every day, and it's different each and every day. Being in 2nd grade, math doesn't come easy. So, your job is going to be to write an application (a program) in C++ that will compute the number of pills that he will take on any given day, based on certain information about his life the day before. Makes sense, right? No, but so what. Every morning, Milhouse will run your program, answering all the questions (inputs), and your program will output the (whole) number of pills he should take.

Specifications: Your program will greet the user, Milhouse. Then, he should be prompted for information necessary to compute his pill dosage for the day. That information is evidenced by the formula needed. The



formulation for computing the number of pills Milhouse will take is very very scientific indeed. Here it is:

of pills = [(# classes studied) / (# hours video gaming) + (soymilk consumed) * (soymilk factor)] +
Tattoo Poisoning (but only IF he had a fake tattoo applied),

where: the number of pills must be rounded *down* every time to ensure no overdose,
#classes is the number of classes (an integer) in school that he had to spend time
studying for the day before - every kid is allergic to school,
#hours video gaming is just that....and an integer value,
soymilk consumed is a (float) value representing the number of cups of soymilk he
drank the day before - yep, he's allergic to soy,
soymilk factor is a constant value known to the medical community which adjusts the
effect of the soy in this calculation; it's current value is 1.9,
Tattoo Poisoning is the pill adjustment factor IF he poisoned his skin with one of those
fake tattoos kids love (this is added in ONLY if he used one the day before), now 3.5.

Note: The value reported by your program should always be rounded down to the nearest integer value. You are **not** allowed to use conditional statements (if or if-else) in the program. Think carefully how to use the information you have input from the user, how it is stored, how boolean variables are represented. It is up to you how you want to name your variables and constants. Don't think that you should copy names that your employer/teacher use to describe a problem. YOU are writing the code, so be sure to use good variable names.

Your program should be "user friendly" in that it should have an opening statement or greeting, user friendly and understandable prompts, and clear and concise outputs and sign-off. Here's a good example of *bad* output:

```
enter stuff: 6000 6 1000 1
5888
```

Notice that there are NO prompts to speak of, and the output is completely unexplained. This is *horrible* output. Something like the following is much much better:

Welcome, Milhouse, to your pill-computer

Please enter the following information:

```
# classes your studied:      3
# hrs gaming:                2
cups soy consumed            1.5
tattoo yesterday? (1 - yes, 0 - no) 1
```

You need to take 7 pills today. Don't choke!

Bye for now, and enjoy your pills.....

Now, isn't that much better?

When you submit: When you submit this, and all subsequent programs for this class, csubmit will compile and run (assuming it compiles) your program during the submission process. Thus, when you submit, you will have to enter inputs as a user of the program. Now, in order to make the output uniform for the grader and to keep them sane, ALL OF YOU will enter the same information. For this assignment, it is

- #classes 3
- #hrs 5



- soy 2.3
- answer in the affirmative

As always, if you have any questions about this assignment, be sure to ask your (lecture) instructor.



E-mail: price@mst.edu | Phone: 573-341-4491 | Fax: 573-341-4501 | Address: 325G Computer Science Building, Rolla, MO 65409-0350

[Report Abuse](#) | [Print Page](#) | [Remove Access](#) | Powered By [Google Sites](#)