

CIS 130 - Intro to Programming - Spring 2007
Homework Assignment #7 - **INDIVIDUAL** assignment

Homework #7 DUE: **BEGINNING** of class, Wednesday, April 11, 2007

Purpose: get practice with simple C++ functions

Note that use of the design recipe is still required for all functions, including C++ functions! But:

- * use C++ **types** in **contracts** for C++ functions, and
- * use `==` for examples for non-void C++ functions
(and, if necessary, note that the C++ **cmath** library, included by the **funct_play2** and **expr_play** tools by default, includes an **abs** (absolute value) function, like Python's, along with other goodies)

For now I will have to just trust that you test these functions thoroughly – at this stage we don't have a convenient way to generate a proof-of-testing file for submission.

ALSO: if you want a **shortcut** for submitting the required homework files, then name them **PRECISELY** using the function names given here, and see the note at the end of this handout.

1. Remember **fahr_to_cels** from HW #1? Write a C++ version of **fahr_to_cels** that takes a single Fahrenheit temperature as its parameter, and returns the equivalent Celsius temperature. No named constants are required for this one, but be sure to test it **carefully**.
2. Remember **dollar_to_euro** from HW #1? Write a C++ version of **dollar_to_euro** that takes a number of U.S. Dollars as its parameter, and returns the equivalent quantity in Euros. Appropriate use of named constants is required for this one.
3. Remember **tank_volume** from HW #2? Write a C++ version of **tank_volume** that takes a tank's length, height, and width as its parameters, and returns the volume of that tank.
4. Remember **semester_grade** from HW #2? Write a C++ version of **semester_grade** that takes a homework average, a quiz average, and a final exam score as its parameters, and returns a semester grade based on the following weighting: 50% from the homework average, 30% from the quiz average, and 20% from the final exam score. Appropriate use of named constants is required.
5. Remember **worked_overtime** from HW #3? Write a C++ version of **worked_overtime** that takes a number of hours worked as its parameter, and returns **bool** true if that number of hours worked is strictly greater than 40, and returns **bool** false otherwise. Appropriate use of **bool** and named constants is required; do not use an **if** in your solution.
6. Write a C++ function **gross_total** that takes a quantity of an item and the cost per item as its parameters, and returns the cost for that many of that item. For this one, your solution should be written so that no fractional item quantities are permitted, but the cost can be fractional.
7. Now write a C++ function **tax_owed** that takes an amount and a tax rate as its parameters, and returns the tax owed for that amount based on that tax rate.
8. And, now write C++ function **final_total** that takes a quantity of an item and a cost per item as its parameters, and returns the total, plus 7.25% tax, for that many of that item. **final_total** must appropriately call **gross_total** and **tax_owed**, and appropriate use of named constants is required.

When you are happy with these functions, you can either submit the **.cpp** and **.h** files for each, OR you can use the following quickie-tool to build a file containing all of them (called a **tar** file – short for **tape archive**...!) and submit that one file instead (IF you have named your functions PRECISELY as given above...):

...if you are interested in the quickie tool, then type the following at the cs-server prompt:

~st10/get_hw07

...give the name of a directory you want built, and when done, if all 16 files are listed on-screen as being in your new file, then you can submit the file whose name it tells you at the end.

(Note: you STILL use ~st10/130submit to submit this homework! But it is your choice if you submit the 16 .cpp and .h files in the usual way, OR use **~st10/get_hw07** and submit the single file it builds containing (hopefully) your 16 .cpp and .h files.)