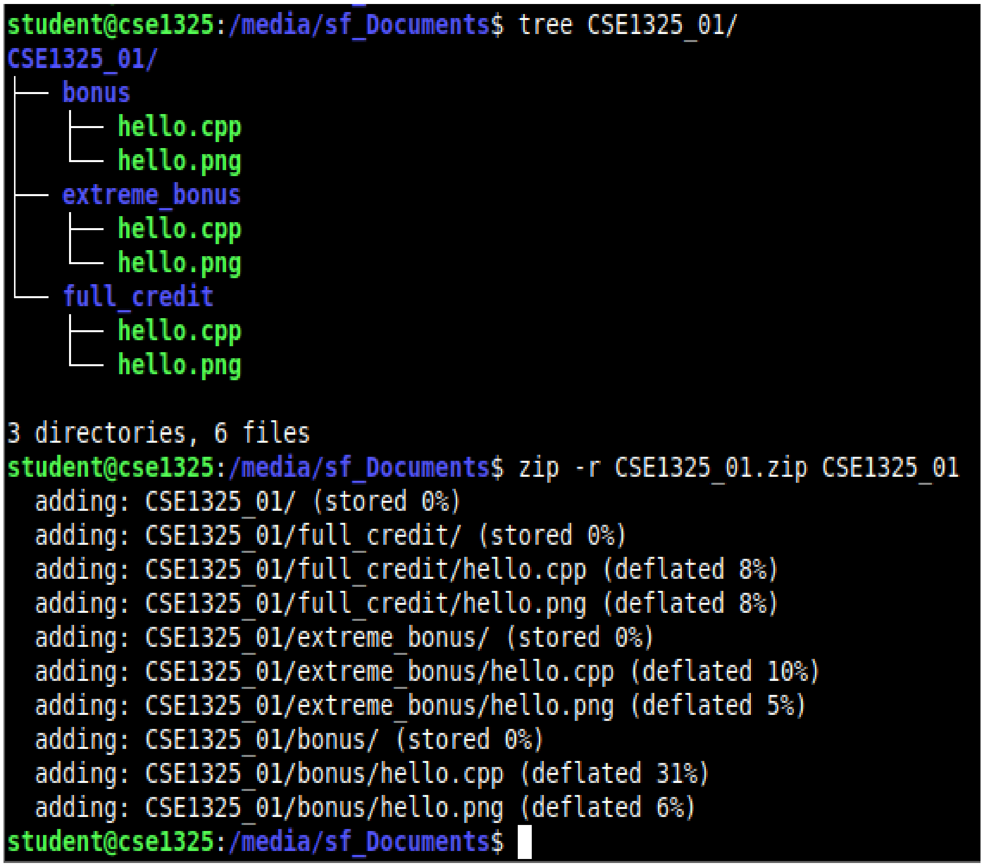
Starting Out – Hello!

CSE 1325 – Spring 2019 – Homework #1
Due Tuesday, January 22 at 8:00 am

Assignment Overview

The classic "first program" in any language simply prints "Hello, World!" to the console. We showed you a some versions of this in class, in several languages. Compiling and running this program is a great way to ensure your environment is operating correctly, and exploring its features.



```
student@cse1325:/media/sf_Documents$ tree CSE1325_01/
CSE1325_01/
├── bonus
│   ├── hello.cpp
│   └── hello.png
├── extreme_bonus
│   ├── hello.cpp
│   └── hello.png
└── full_credit
    ├── hello.cpp
    └── hello.png

3 directories, 6 files
student@cse1325:/media/sf_Documents$ zip -r CSE1325_01.zip CSE1325_01
  adding: CSE1325_01/ (stored 0%)
  adding: CSE1325_01/full_credit/ (stored 0%)
  adding: CSE1325_01/full_credit/hello.cpp (deflated 8%)
  adding: CSE1325_01/full_credit/hello.png (deflated 8%)
  adding: CSE1325_01/extreme_bonus/ (stored 0%)
  adding: CSE1325_01/extreme_bonus/hello.cpp (deflated 10%)
  adding: CSE1325_01/extreme_bonus/hello.png (deflated 5%)
  adding: CSE1325_01/bonus/ (stored 0%)
  adding: CSE1325_01/bonus/hello.cpp (deflated 31%)
  adding: CSE1325_01/bonus/hello.png (deflated 6%)
student@cse1325:/media/sf_Documents$ 
```

Full Credit: **Under Ubuntu Linux 18.04**, create and run the C++ "Hello, World!" program using cout, **replacing "World" with *your name***. Deliver a **single ZIP archive file named CSE1325_01.zip** via Blackboard, containing your **source file named hello.cpp** and **a screenshot named hello.png** of your program in action, using the exact file hierarchy shown**.**

Bonus: Instead of hard-coding your name, **ask the user for their name using cin**. Then **print "Hello, [name]"**, where [name] is whatever name they enter. Add a **"bonus" subdirectory** to CSE1325_01 with your **source file named hello.cpp** and one or more **screenshots named hello.png, hello1.png, hello2.png, …** showing your program in action with various inputs (multiple word names, Unicode characters, and so on)**.**

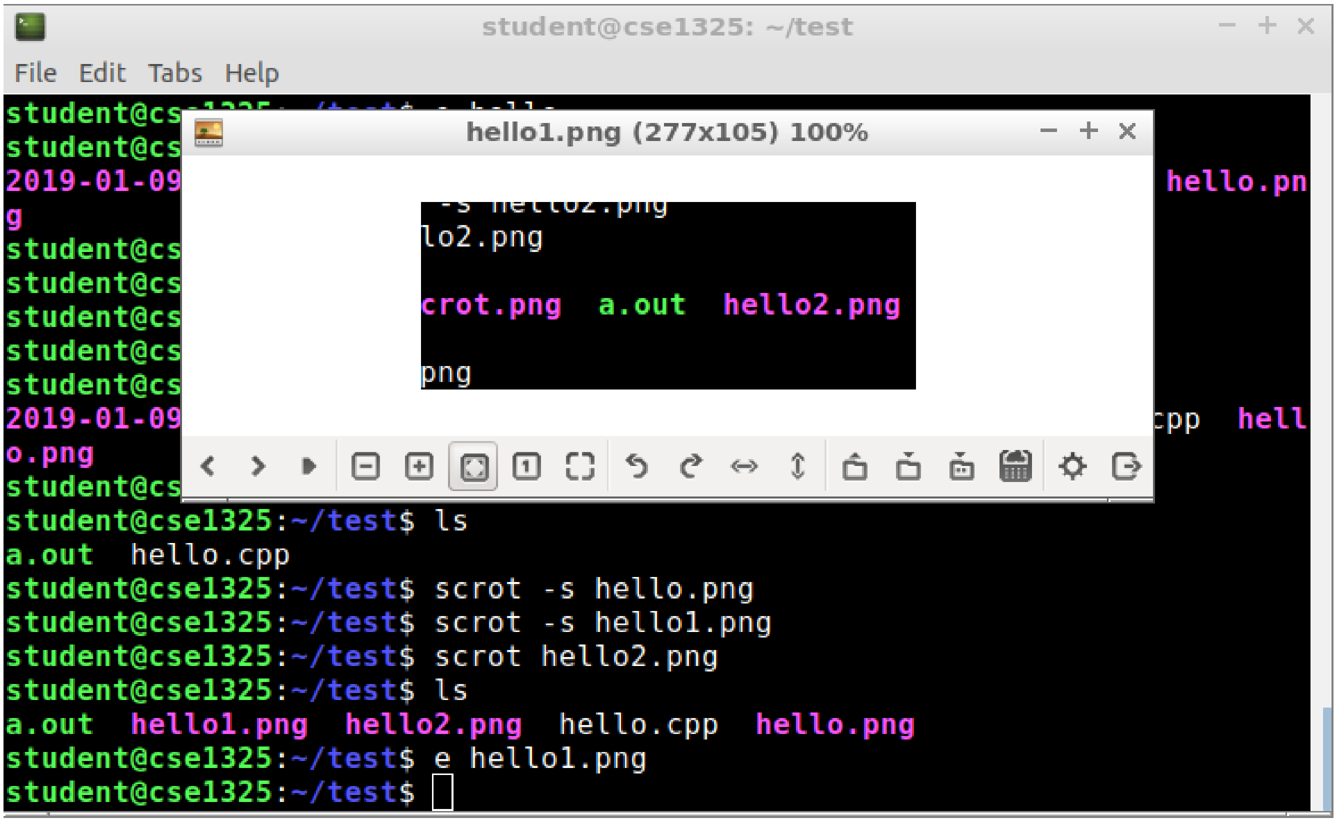
Extreme Bonus: Instead of asking the user for their name, **automatically identify the current user and print "Hello, [name]!"** ***without any input or hard-coded names in the source code***,where "[name]" is *something that uniquely identifies the user*, such as a user name, proper name, etc. Add an **"extreme_bonus" subdirectory** to CSE1325_01.zip with your **source file named hello.cpp** and one or more **screenshots named hello1.png, hello2.png,** etc. showing your program in action. For maximum credit, test the program in at least two user accounts and on at least two operating systems.

Additional information that you may find helpful follows.

Screenshots

Lubuntu 18.04 (part of the VM appliance) includes a program called **scrot** (for **scr**eensh**ot**). Type "**man scrot**" for the manual, using PgUp and PgDn or your mouse wheel to explore, and "q" to quit.

- **To capture a single window** to a file name (ahem) hello.png, e.g., your terminal window, type "**scrot -s hello.png**", then click the terminal window with your mouse. The "-s" means "select" what to capture. You can view the screenshot using "**e hello.png**".
- **To capture a rectangular area** from the screen to file hello1.png, type "**scrot -s hello1.png**", then move the mouse to the upper left corner of the rectangle you want to capture, hold down the primary mouse button, drag the mouse to the lower right corner, and release.
- To **capture the entire screen** to hello2.png, simply type "**scrot hello2.png**". Done and done.



```
-s hello2.png
lo2.png

crot.png  a.out  hello2.png

png
```

```
student@cs
student@cs
2019-01-09                                                          hello.pn
g
student@cs
student@cs
student@cs
student@cs
student@cs
2019-01-09                                                        cpp   hell
o.png
student@cs
student@cse1325:~/test$ ls
a.out  hello.cpp
student@cse1325:~/test$ scrot -s hello.png
student@cse1325:~/test$ scrot -s hello1.png
student@cse1325:~/test$ scrot hello2.png
student@cse1325:~/test$ ls
a.out  hello1.png  hello2.png  hello.cpp  hello.png
student@cse1325:~/test$ e hello1.png
student@cse1325:~/test$ 
```

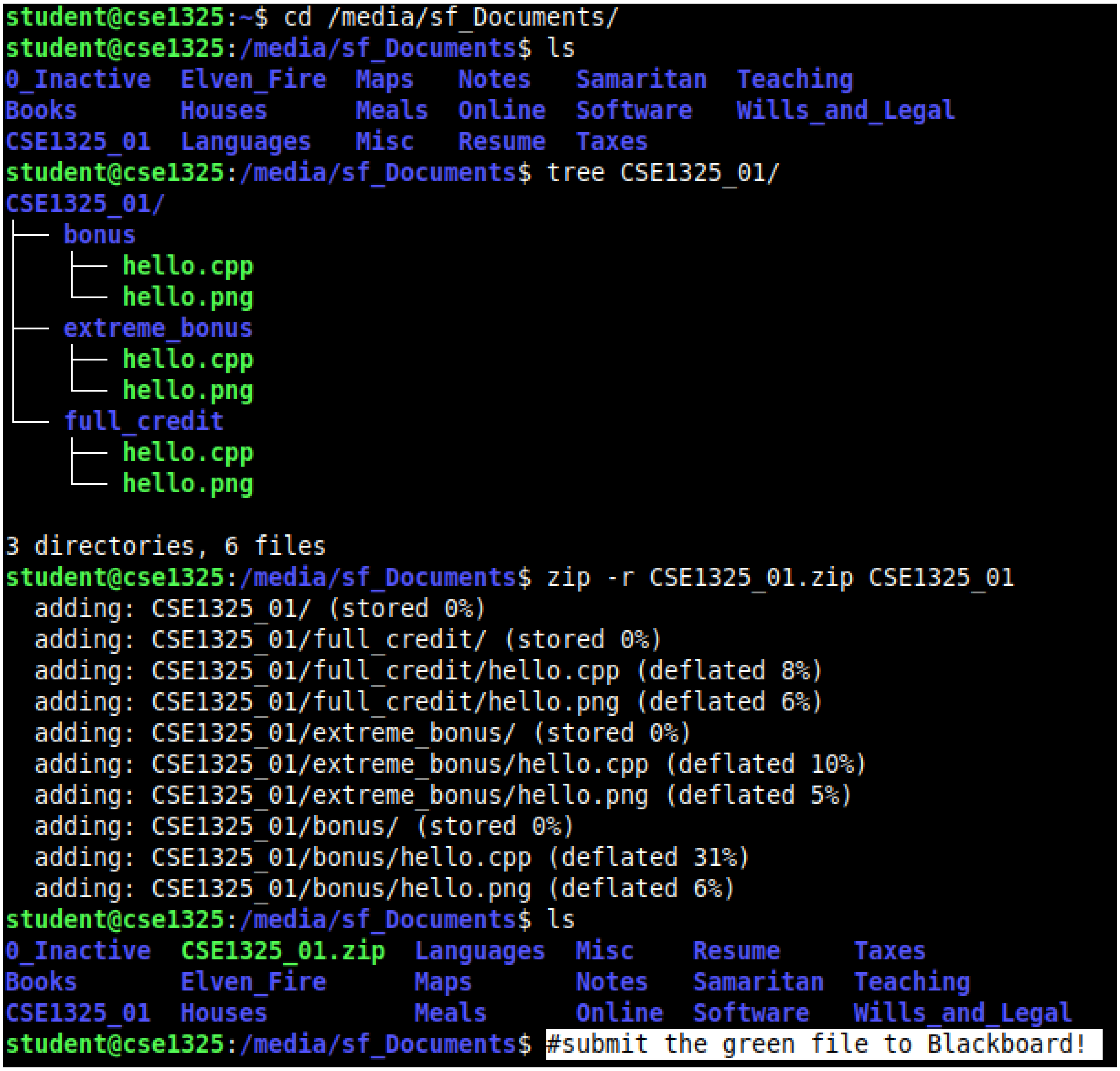
If you installed your own Linux operating system and scrot is missing, simply install scrot using "**sudo apt install scrot**". Of course! (You may use any screenshot utility you prefer, though, as long as your screenshots are delivered in PNG format.)

ZIP Archives

An “archive” is a file that can contain entire directories of files. The most common are called ZIP, which is not an acronym – it was chosen to imply that it was faster than other archive formats. 😄

To **create a ZIP archive in bash**, change to the directory containing your CSE1325_01 directory (this may be “**cd /media/sf_Documents**”), then create your archive using “**zip -r CSE1325_01.zip CSE1325_01**” (the -r means **recursive**, that is, include all the files in all of the subdirectories as far down as they go). Notice the archive name is first on the line, then list directories and files to put in it.

You can extract the files from a ZIP archive by using (wait for it) “unzip CSE1325_01.zip”, and list the contents of a ZIP archive with “unzip -l CSE1325_01.zip” (the -l means list – that’s an el, not a one).



```
student@cse1325:~$ cd /media/sf_Documents/
student@cse1325:/media/sf_Documents$ ls
0_Inactive  Elven_Fire  Maps   Notes   Samaritan  Teaching
Books       Houses      Meals  Online  Software   Wills_and_Legal
CSE1325_01  Languages   Misc   Resume  Taxes
student@cse1325:/media/sf_Documents$ tree CSE1325_01/
CSE1325_01/
├── bonus
│   ├── hello.cpp
│   └── hello.png
├── extreme_bonus
│   ├── hello.cpp
│   └── hello.png
└── full_credit
    ├── hello.cpp
    └── hello.png

3 directories, 6 files
student@cse1325:/media/sf_Documents$ zip -r CSE1325_01.zip CSE1325_01
  adding: CSE1325_01/ (stored 0%)
  adding: CSE1325_01/full_credit/ (stored 0%)
  adding: CSE1325_01/full_credit/hello.cpp (deflated 8%)
  adding: CSE1325_01/full_credit/hello.png (deflated 6%)
  adding: CSE1325_01/extreme_bonus/ (stored 0%)
  adding: CSE1325_01/extreme_bonus/hello.cpp (deflated 10%)
  adding: CSE1325_01/extreme_bonus/hello.png (deflated 5%)
  adding: CSE1325_01/bonus/ (stored 0%)
  adding: CSE1325_01/bonus/hello.cpp (deflated 31%)
  adding: CSE1325_01/bonus/hello.png (deflated 6%)
student@cse1325:/media/sf_Documents$ ls
0_Inactive  CSE1325_01.zip  Languages  Misc    Resume     Taxes
Books       Elven_Fire      Maps       Notes   Samaritan  Teaching
CSE1325_01  Houses          Meals      Online  Software   Wills_and_Legal
student@cse1325:/media/sf_Documents$ #submit the green file to Blackboard!
```

(Note: “tree” is NOT pre-installed in the appliance. Install it with “**sudo apt install tree**” if you like.)

Deliverables

You will deliver a single ZIP archive file named CSE1325_01.zip via Blackboard (subsequent assignments will be CSE1325_02.zip, etc.). This archive will contain a directory named CSE1325_01, which will then contain one subdirectory for each level of the homework you attempt: full_credit, bonus, and extreme_bonus (*spelling matters* – use the underscores!). **Each subdirectory will contain at a minimum a source file named hello.cpp and a screenshot named hello.png.**

You *may* use git for the first homework – it's a great opportunity to practice - but you won't lose points if you don't. If you use it incorrectly, the graders will provide feedback, giving you a better chance at getting it right for the second homework where it is required. **To try completing assignment 1 with git, follow the step-by-step instructions in Section 6, "Submitting your homework with a git repository" in the "Git in 5 Pages" document on Blackboard.**

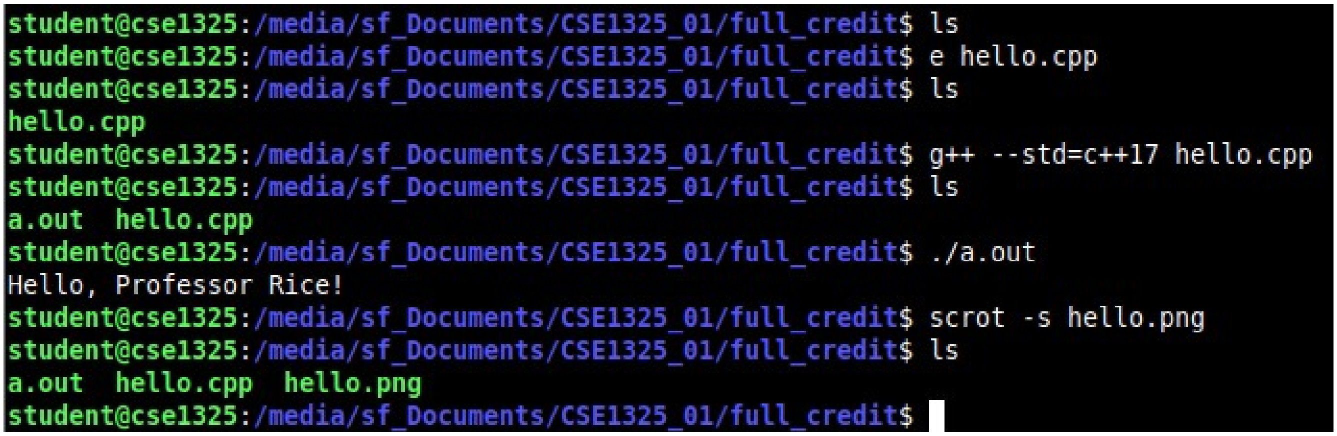
Full Credit

Set up Linux with your preferred editor (gedit is recommended) and the g++ 7.3 compiler environment as discussed in class and the slides; the appliance is ready to go. Create the Hello, World! program just as we did in class (the appliance has a pre-typed copy at /home/student/Documents/Examples/hello), and verify that it compiles and runs ("**g++ --std=c++17 hello.cpp**" then "**./a.out**"). Explore your new environment, referencing "**Bash in 5 Pages**", until you are comfortable with it.

Now edit the program to replace World with *your name*. Thus, if your name is "Professor Rice", your program would print "Hello, Professor Rice!" (and you would be teaching this class).

In the full_credit directory, you will provide a file called hello.cpp and a screenshot named hello.png (in Portable Network Graphics format) of your program **running and printing *your name*** from the Ubuntu terminal. For example, if your name is Hien, your program would print "Hello, Hien!"

Completing this task earns 100 points. **This must be completed prior to attempting any of the bonus levels.** You will receive no credit for bonus work if you do not complete the full credit assignment first.



```
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ ls
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ e hello.cpp
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ ls
hello.cpp
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ g++ --std=c++17 hello.cpp
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ ls
a.out  hello.cpp
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ ./a.out
Hello, Professor Rice!
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ scrot -s hello.png
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ ls
a.out  hello.cpp  hello.png
student@cse1325:/media/sf_Documents/CSE1325_01/full_credit$ 
```

Bonus

In the “bonus” directory, you will provide a second file also named hello.cpp. Instead of hard-coding your name, **ask the user for their name. Then print “Hello, [name]”**, where [name] is whatever name they enter. (Hint: Try “cin >> name;” to read a string from the keyboard into variable “name”. Input is covered in more detail in Lecture 02.)

Test the program to identify any limits on your system – for example, can your program handle Unicode characters such as José, Paŝtisto, or 名 ? Can it handle names with spaces such as “Professor Rice”? Can you paste a name at your prompt, and copy the output? (Note: Your program isn't *required* to do all of these things, but your testing should clearly demonstrate whether it can or not.)

For credit, you must deliver hello.cpp and *one or more* screenshots (named hello.png, hello1.png, hello2.png, etc.) showing *significant* (in the grader’s opinion) testing – **the more significant your testing, the more valuable the bonus.** It’s fine, in fact recommended, to include many tests in a single screenshot.

Completing this task earns *up to* an additional 10 points.

Extreme Bonus

Your environment likely knows quite a bit about you, including at least your user name and possibly your full name. Let's automate our greeting. (Note: We will NOT cover the C++ instructions needed to do this task – hence “extreme”. Google, Stack Overflow, et. al. are your friends.

Begin learning how to learn how to program now!

In the “extreme_bonus” directory, create a third program called hello.cpp that **automatically identifies the current user, and prints “Hello, [name]!”** ***without any input or hard-coded names in the source code*** (where “[name]” is something that uniquely identifies the user, such as a user name, proper name, user identification (UID), etc.).

Test the program **in at least two user accounts**, and (if possible) **on multiple operating systems**, capturing screenshots as hello.png, hello1.png, hello2.png, etc. Is your program portable? What other information can you obtain about the current user?

Completing this task earns *up to* 15 additional points, depending on how well you impress the grader.

More on This (and Future) Homework

All homework must be turned in via Blackboard by the deadline. **Late submissions are never graded** without an *exceptionally* good excuse, but the grader will grade the latest submission prior to the deadline.

Start and finish early. It's a great habit to develop for life. **You may submit archives as many times as you like**, and we will only grade your latest submission that is prior to the deadline. Don't accept a 0 after completing 90% of the work! Thus, **if you are concerned that you won't finish an assignment, go ahead and submit partial work.** Then keep working. If you finish successfully and submit by the deadline, we'll simply ignore your earlier submissions.

It is possible – indeed, rather easy – to earn more than a 100 on homework assignments. Going beyond the requirements for "Full Credit" will help you to grow your skill, and also result in bonus points. If you have more than a 100 average on your homework at midterm or the end of the class, *the additional credit will be included in calculating your final average*. This is a ***GREAT*** way to recover from the occasional missed exam question.

Please do not wait until the end of the semester to seek "extra credit work" to pass the class or improve your GPA. Extra credit work is *right here*! Start today and avoid the rush.