Course Overview

CSC230: C and Software Tools

N.C. State Department of Computer Science

What will we do in here?

Your instructors

- Professor: **Tyler Bletsch, PhD**
  - tkbletsc@ncsu.edu
  - Office Hours: by appointment (but ENCOURAGED)
- TA: **Brantley Collins**
  - bjcollin@ncsu.edu
  - Office Hours: TBD
- Lecture aide: **mchammer.gif**
  -

  - Points out especially important facts.

Webpage

- http://courses.ncsu.edu/csc230/lec/051/

  (You can get there via WolfWare, too.)

  A Moodle-free course.

- Lecture notes and homeworks are linked from the schedule table.

Gradebook

- We'll be using Wolfware Classic Gradebook
- Access via:
  - – http://courses.ncsu.edu/csc230 and select the Gradebook link for your section
- NOTE: The final average and letter grade shown should be assumed to be INACCURATE until the last assignment has been input.

Getting Help

Email: csc230-051-sup@wolfware.ncsu.edu

See Course Website for

Instructor and TA office hours!

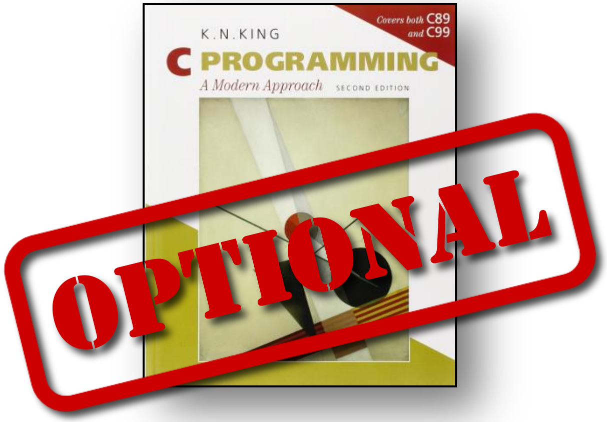
Ask general questions on the course forums

(Piazza site linked from course webpage)

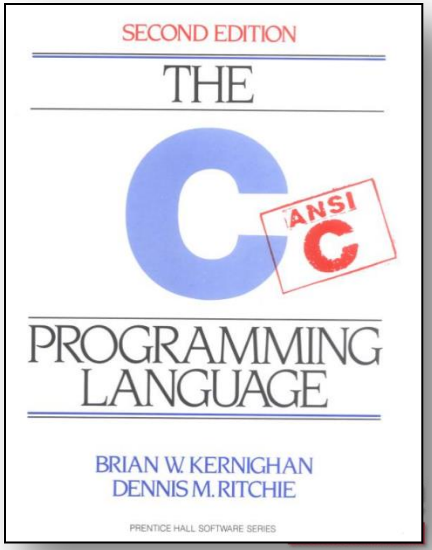
No code!

The Textbook

- King's *C Programming: A Modern Approach, 2nd ed.*



- Alternative: Kernighan and Ritchie. The C Programming Language, 2nd ed.
  - Commonly referred to as "K&R".



If you go to **addall.com**, you can search all online booksellers at once.
Amazon price for text: $83 used. Addall found it for $64.
Amazon price for K&R: $24. Addall: $12.

Grading Breakdown

Assignment	%
Homeworks	50%
Exercises and Attendance	5%
Exam 1	15%
Exam 2	15%
Exam 3	15%

Minimum Grade Requirement

In order to receive a final grade of D- or higher, you must have an average of 50% or higher on all three exams and an average of 50% or higher on all of the Homeworks. Students failing to meet these requirements will receive a grade of F for the course.

Homeworks

- 6 homeworks during the summer
  - Lowest homework grade dropped *provided you submit a reasonable attempt at each one (no blowing one off)*
- Most weeks have homework
  - Those weeks without homework have exams
- Late homeworks accepted up to 24 hours after the deadline
  - Automatic 20 point deduction for late homeworks
- All homeworks are due at 11:45pm on the assigned deadline
- No homeworks will be accepted via email

Exercises

- Most classes will have exercises
  - Bring your laptops/tablets to class!
  - Most exercises will be done via a Google form.
  - The exercises for a week will be due on Sunday at 11:45pm. To get credit, you must submit at least once within an hour of class ending, but we'll grade your latest submission.
  - No exercises will be accepted late

Exams

- All exams are cumulative up to the lecture before the exam period
- There are no homework deadlines on a week containing an exam
- The final exam is cumulative and held during the standard final exam period
- See the course website for exam dates

Grade Appeals

- All regrade requests must be in writing
  - Email the sup list with your questions
- The TA that graded your assignment will respond first
- If you still have concerns, contact the instructor

- All regrade requests must be submitted to the instructor no later than 2 weeks after the assignment was returned to you.

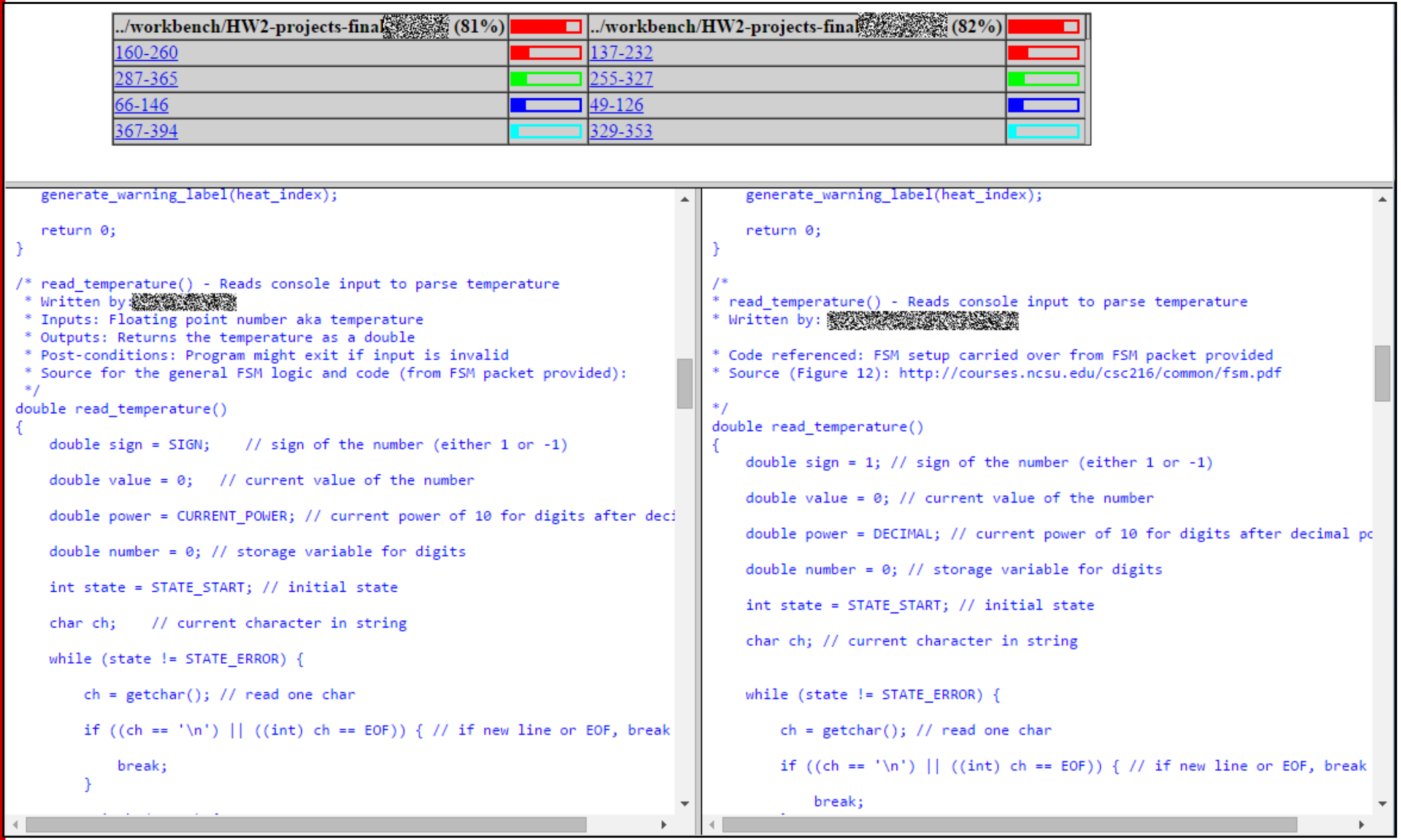
Disclaimer: A Note About My Code

- Code shown in lecture slides is incomplete. It…
  - may not compile
  - may compile but give warnings
  - may not check the input for possible errors
  - may not check return values for errors
  - may not be secure against deliberate exploits
  - may be inefficient
  - may not have many comments or be formatted well
- It is NOT OK for your programs to have these problems!

Do your own work!!

- All work that you turn in for grading must be your own!
  - Unless stated otherwise in the assignment
- Students who cheat on a homework, exercise, or exam will receive a -100% for the assignment!!!
- What about…
  - Code in the book? 🙂
  - Code from lecture? 🙂
  - Code on websites or other books? 🚫
  - Code from fellow students? 🚫
  - Hard coding test solutions? 🚫
  - Circumventing the automated grading? 🚫

MOSS:
Measure of Software Similarity



../workbench/HW2-projects-final[redacted] (81%)	🟥	../workbench/HW2-projects-final[redacted] (82%)	🟥
160-260	🟥	137-232	🟥
287-365	🟩	255-327	🟩
66-146	🟦	49-126	🟦
367-394	🟦	329-353	🟦

```c
    generate_warning_label(heat_index);

    return 0;
}

/* read_temperature() - Reads console input to parse temperature
 * Written by:[redacted]
 * Inputs: Floating point number aka temperature
 * Outputs: Returns the temperature as a double
 * Post-conditions: Program might exit if input is invalid
 * Source for the general FSM logic and code (from FSM packet provided):
 */
double read_temperature()
{
     double sign = SIGN;    // sign of the number (either 1 or -1)

     double value = 0;   // current value of the number

     double power = CURRENT_POWER; // current power of 10 for digits after deci

     double number = 0; // storage variable for digits

     int state = STATE_START; // initial state

     char ch;    // current character in string

     while (state != STATE_ERROR) {

         ch = getchar(); // read one char

         if ((ch == '\n') || ((int) ch == EOF)) { // if new line or EOF, break

             break;
         }
```

```c
    generate_warning_label(heat_index);

    return 0;
}

/*
* read_temperature() - Reads console input to parse temperature
* Written by: [redacted]

* Code referenced: FSM setup carried over from FSM packet provided
* Source (Figure 12): http://courses.ncsu.edu/csc216/common/fsm.pdf

*/
double read_temperature()
{
    double sign = 1; // sign of the number (either 1 or -1)

    double value = 0; // current value of the number

    double power = DECIMAL; // current power of 10 for digits after decimal pc

    double number = 0; // storage variable for digits

    int state = STATE_START; // initial state

    char ch; // current character in string


    while (state != STATE_ERROR) {

        ch = getchar(); // read one char

        if ((ch == '\n') || ((int) ch == EOF)) { // if new line or EOF, break

            break;
```

Doesn’t care about:

- Comments
- Whitespace
- Naming
- Values

Only cares about code structure.

<u>How to beat it?</u>

Write your own code

Our Responsibilities

- The instructor and TA will…
  - Provide lectures at the stated times
  - Set clear policies on grading
  - Provide timely feedback on assignments
  - Be available out of class to provide reasonable assistance
  - Respond to comments or complaints about the instruction provided

Our Responsibilities

- Students are expected to…
  - Receive lectures at the stated times
  - Turn in assignments on time
  - Seek out of class assistance in a timely manner if needed
  - Provide frank comments about the instruction or grading as soon as possible if there are issues
  - Assist each other *within the bounds of academic integrity*

QUESTIONS? COMMENTS?