

UNC Charlotte, Department of Electrical and Computer Engineering
ECGE 3/5/6090, Fall 2003, Homework #6, Due: 10/8/03, at the beginning of class (20 points)
Turn in hard copy in class AND send assignment (MS Word doc or text) to jmconrad@uncc.edu

It should be obvious, but you may NOT use questions from previous tests or that have been submitted by other students from other semesters (including this semester)!!!!

Part A: Write three multiple-choice questions suitable for the midterm exam. The requirements:

1. They must be typed.
2. There must be five choices for answers.
3. Only one answer should be possible. Identify the answer by “**bolding**” it.
4. Ensure the questions and answer-choices are clear.

The scoring for each question will be:

- 1 point: Clear question, from material covered since the start of semester
- 1 point: Clear and reasonable answer-choices, answer correct and **bolded**
- 1 point: Quality of question and answer-choices (spelling, grammar, adherence to class material)
- 1 point: Difficulty of question (determined by Dr. Conrad)

Hints:

- You can include a code listing, and ask several questions related to the code listing.
- You can provide several supporting lines or identify a problem, and then ask several questions related to the problem you set up.
- ENSURE there is only one correct answer.

Part B: Write two "long or short" answer question, something that requires design and thought

The scoring the question will be:

- 1 point: Clear question, from material covered since the beginning of the semester.
- 1 point: Answer to question provided.
- 1 point: Quality of question and answer (spelling, grammar, adherence to class material)
- 1 point: Difficulty of question (determined by Dr. Conrad)

For example: Consider the following C code

```
//-----  
// This ISR handles an external hardware interrupt INT3. * It reads a byte from port 1, and stores it in  
// a circular buffer, data_in It also removes a byte from the circular buffer, data_out and writes it to port 3  
// data_in_ptr indexes the next free location in data_in  
// data_out_ptr indexes the next byte to send  
//-----  
unsigned char data_in[CIRC_BUF_SIZE], data_out{CIRC_BUF_SIZE};  
unsigned char data_in_ptr=0, data_out_ptr=0;  
  
#pragma INTERRUPT Transfer_ISR  
void Transfer_ISR() {  
    data_in[data_in_ptr] = p1;  
    data_in_ptr++;  
    data_in_ptr %= CIRC_BUF_SIZE;  
    p3 = data_out[data_out_ptr];  
    data_out_ptr++;  
    data_out_ptr %= CIRC_BUF_SIZE;  
}
```

Write the assembler directives needed in sect30.inc to link INT3 to your ISR.

```
.glob _Transfer_ISR  
.lword _Transfer_ISR
```