From Homework, Week 4
CSC 302. Covered in class before the midterm exam. Only three or four words changed to fit one page for printing.

26. Explain why the "user interface" to a safety-critical software system may add risk of accidents to a system. Is the proper approach to such problems "RTFM," human-factors analysis (industrial psychology and HCI work), or a combination of these?

Approach to answer a question:

a. Facts, definitions needed for analysis
b. Restate the question you will answer
c. Note alternative possibilities (and sources if possible) - or state a well known position.
d. Analyze to come to a conclusion. Be brief, concise, use simple language and simple sentences. Have good, solid reasoning behind the conclusions. A simple example to illustrate reasoning is a very good device. (Making assumptions is weak, if you need an assumption to proceed, state and justify it in the first section with other "facts.")

This is from homework question number two. TWO parts to the question, separate out the parts and proceed.

USER INTERFACE CAN ADD RISK:

The "user interface" to a computer system is the only way in which the human operator or user can interact with the system to get the desired result. [Turner] Safety-critical software is software that controls a system that may cause injury to humans. [Baase]

Can the "user interface" add risk of accidents to a system (that was not otherwise as risky?)

[Baase] states several factors critical to a good user interface:
- clear instructions and error messages
- consistency
- checking of appropriate input to avoid failures due to typos or other silly human errors

If any of these factors are not well considered during user interface design and implementation, the operator or user of a safety-critical system could easily make a mistake that leads to a dangerous failure. Consider the pilot of a modern airliner. If the front panel display interface does not provide clear instructions during an otherwise confusing or stressful situation (like aggressive stewards wanting help with a drunk passenger), the pilot is likely to make a quick decision on poor information. This is likely to increase risk to the safety of the plane.

PROPER APPROACH:

Designers, implementors and administrators of software user interfaces use several approaches to ensuring proper operation of the system. RTFM means "read the fu__ing manual" and represents the techno-geek's approach that the human is responsible to learn esoteric commands and adjust their human behavior to the "needs" of the computer. [Turner] An alternative approach is that of the industrial psychologists and human factors analysts who would like the computer software to be designed to work with humans on their own terms (even with human errors.) [Norman]

What is the proper approach to reduce the risk possibly added by the user interface to a safety-critical system?

The RTFM approach assumes that the operator or user will take the time to receive training and will concentrate on how to interface with the machine. Humans need to learn how to properly interface with a serious machine that has a safety-critical interface. However, designing the interface to account for human traits such as error-proneness might be really helpful in the safety-critical realm as it takes into account the reality of use of human operators and users. This might be more difficult for the software interface designer, she would need to learn some basic psychological principles of human behavior and make that part of the technical specifications.

It is known that humans make errors in following esoteric instructions. [Norman] This can increase risk of error and failure. For RTFM, if manuals are well written and error codes clearly explain what needs to be done, this can possibly minimize the risk of problems in an environment where decisive human action is not required. In safety-critical situations, we cannot count on a lot of time for deliberation. Taking into account the reality of human behavior when the human is required to interface with the software makes sense. After all, the user interface is "designed" to help the user interact with the computer system, it should not be the other way around. This is especially true when lives are at stake and simple human errors are well understood.

I recommend a combination of clear manuals, training, and user interfaces designed to work with the human users.