CSCI 255: Introduction to Embedded Systems
Fall 2011

Course Objective:

Develops basic concepts of computer systems and computer architecture. Includes base-2 arithmetic, octal and hexadecimal number systems, computer addressing modes, I/O, assemblers. Prerequisites: CSCI 111 or CSCI 112 or consent.

Instructor

Keith Vertanen
kvertanen@mtech.edu
Museum 102, 496-4385

Office hours:	Tuesday	4:00 – 5:00pm
	Thursday	4:00 – 5:00pm
	Friday	3:00 – 4:00pm
	or by appointment	

Classes

Mon	02:00 - 04:50pm	Lab	NRB 228
Wed	02:00 - 02:50pm	Lecture	CBB 112
Fri	02:00 - 02:50pm	Lecture	CBB 112

Resources

Textbook	The 8051/8052 Microcontroller by Craig Steiner
Class web page	http://katie.mtech.edu/classes/csci255/
Moodle	https://moodlemtech.mrooms3.net/course/view.php?idnumber=72253

Evaluation

A	90% - 100%	Midterm exam	20%
B	80% - 89%	Final exam	30%
C	70% - 79%	Assignments	50%
D	60% - 69%	Staff discretion (participation and extra-credit)	±?%
F	0% - 59%		

Assignments

Assignments will consist of both written exercises and hardware/software assignments. The exact details of how and when to submit will be stated in the assignment description. You get a total of four free late days. Each late day buys you a 24-hour extension to an assignment submission deadline. If you are out of free late days, any further late submissions will be given a zero. In the case of late assignments that require you to demonstrate functioning hardware, it is up to you to schedule an appointment with the instructor or TA.

Your programs will be graded on correctness, programming style (including comments), and efficiency. Partial credit is possible so if you run out of time, submit what you have.

Exams

There will be two exams that cover material from lectures, labs, and assignments. Makeup exams will only be given if you bring valid documentation explaining a legitimate reason for missing the test. The midterm exam will occur during one of our scheduled lab periods. The final exam will be comprehensive.

Honor code policy

Cheating will not be tolerated and can result in failure of the course. Submitted work must be your own. Under no circumstances should you copy another person's solution or code. Exams are to be strictly your own effort. No electronic devices are allowed in exams.

Programming is a creative process and no two programmers will solve the same problem in the same way. You are encouraged to discuss how to design a solution to a given problem with your classmates. But when it comes time to convert your design into code, you must write the code yourself. Be sure not to leave copies of your code where others might be able to access it (such as in the recycling bin of a lab computer). You may adapt code from the course materials provided you cite what code you used in your program's comments.

General

Any student who may need an accommodation due to a disability, please make an appointment to see me during my office hours. A letter from a Montana Tech Disability Coordinator authorizing your accommodations is needed.

Expectations

E1. Students have a high-level understanding of the fundamental operations of a computer. (CSCI 111, R1)

E2. Students can use an editor and a compiler or interpreter to design, write and execute programs in a high-level programming language that comply with the MTech CS design language and programming language standards. (CSCI 111, R3)

E3. Students understand data types, variables, assignment, arithmetic and boolean expressions. (CSCI 111, R4)

E4. Students know how to use the basic selection and repetition control structures in a high-level programming language. (CACI 111, R5)

Course Outcomes

R1. Students understand basic digital circuits including transistors, logic gates, and latches. (CAC-c; EAC-c)

R2. Students understand at a high-level how a computer processor operates (fetch-and-execute cycle, interrupts, registers, memory, addressing, etc.) (CAC-a; EAC-a)

R3. Students know and can use numbers in any base and can convert numbers between bases. (CAC-a; EAC-a)

R4. Students can use an editor and an assembler to write and execute assembly and C language programs for a specific processor and Software Development Kit. (CAC-c, i; EAC-c, k,1, 3)

R5. Students know the assembly language instruction set for a specific processor. (CAC-c: EAC-1)