CS3432 Learning Outcomes

family	**Prerequisite knowledge from CS1, CS2, digital design, discrete, and precalc**	**Students will be familiar with...**	**Students will be able to effectively apply skills...**	**Students will be able to analyze & synthesize solutions…**
NR: numeric representation & ops	- familiar with radixes, signed representations, and scientific notation		- convert hex, decimal, signed-decimal, binary binary metric - signed/unsigned comparison (flags, order) - Add-with-carry - cast/sign-extend - floating point	- can determine appropriate representations for elementary types and design low-level programs that compute arithmetic results
L: linearization	- algebra, - arithmetic & control-flow structures of an oo language - block structures,		- expressions (incl side effects) - control flow (if/while/for) - translate boolean logic - branch tables - op on arrays, structs, and pointers	- can translate infix expressions and block-structured programming constructs to assembly language
GA: gross architecture	- able to program in at least one oo language - familiar with combinational and sequential logic	Can describe the fetch-execute cycle in the context of the roles of PC, SP, flags, registers and memory.	- select appropriate instructions - specify operand order - encode and decode instructions - specify addressing mode - utilize interrupt mechanism - implement interrupt handlers	- can implement and debug simple imperative programs in assembly or machine language

Ti: timing	- algebra - synchronous logic - frequency		- determine cycles/instruction - determine which instructions repeat in a loop	- can compute the execution time of a simple loop - can design a loop that delays a specified amount of time
Sub: subroutine linkage & separate compilation	- in oo languages studied in CS1/2		- parameter passing - return value - allocation of auto vars - register usage - global/local symbols	- can write or call a method with local variables, parameters, and return value in assembly language
VA: variable allocation	- in oo languages studied in CS1/2		Can define and use variables with various.. - scope: visibility (file/method/program) - variable lifetime (program/method) - size - alignment - arrays - pointers - structs	Can appropriately allocate static and auto variables including arrays and pointers in assembly language
T: tools	- ide - hierarchical filesystems		Can effectively employ in the composition and debugging of programs - editor - compiler - make - bash - gdb - source code repo	Can compose and debug simple programs in a command-line environment

WC: written communication	- proficient in English		Can - interpret technical documentation on familiar topics - describe implementations that they design - recognize/use technical terminology - appropriate documentation for code	Can appropriately document simple programs
MP: mature programming	- proficient in OO programming - appropriate comments - can modularize - appropriate symbol names - coding style		Can utilize in a program - appropriate comments - modularize - imperative programming - appropriate symbol names - coding styles	Can appropriately modularize and document simple programs consisting of multiple files
Perf: advanced topics for performance		- pipelining - vectorization - predicated instructions		Can identify when these topics are relevant to constructing an efficient solution.
DEV: devices	gates, latches, (de)multiplexers, ALUs, switches, counters	- gross characteristics of memory & storage devices - counter-timer	can implement - simple programmed i/o - interrupt handlers	Can design programs that implement simple programmed i/o and interrupt handling - can determine the types of storage devices suitable for a variety of uses.