# TCSS 371 Master Syllabus
# (Approved: June 2, 2017)
# (Effective: Autumn 2017, with prereq change to TCSS 321 to take place as of Autumn 2018)

**Course Title**

Machine Organization

**Catalog Description**

Develops the hardware basis for computing systems, and the relationship between hardware and software. Covers number representations, digital logic, machine organization, instruction set architecture, assembly language, and translation of high-level languages into machine instructions. Prerequisite: a minimum grade of 2.0 in TCSS 333 or equivalent.

**Preconditions**

Prior to taking the class, student must:

- Know how to program using a high-level language
- Understand logic operations such as "and," "or," "xor," and "not."
- Be able to use an integrated development environment (IDE).
- Understand simple programming data structures.

**Course Objectives**

The objectives of this course are to teach students:

- the fundamentals of computer organization (CPU, Memory and I/O) and Instruction Set Architecture.
- digital logic related to implementation of the above.
- the basic machine cycle and computational operations.
- essential features of assembly language.

**Student Learning Outcomes**

Upon successful completion of the course, students should be able to:

- convert numbers between various bases and use two's complement
- explain the function of basic digital logic circuits
- design simple logic circuits after specifying their behavior using a truth table
- translate between assembly instructions and machine code
- explain the instruction execution cycle
- write small programs in assembly language including function calls

**Relationship of course to CSS student learning outcomes:** This course supports and assesses the achievement of the following elements of the program objectives:

- An ability to apply knowledge of computing and mathematics appropriate to the discipline
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- An ability to use current techniques, skills, and tools necessary for computing practice.

**Relationship to UWT student learning goals** (to be added to syllabus handed out to students)

- *Inquiry and Critical Thinking*: Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.
- *Communication/Self-Expression*: Students will gain experience with oral, written, symbolic and artistic forms of communication and the ability to communicate with diverse audiences. They will also have the opportunity to increase their understanding of communication through collaboration with others to solve problems or advance knowledge.

**Topics Covered:**

- **Digital Logic**
  - Analog and digital systems
  - Logic gates, gate delays and flip-flops
  - Boolean logic**,** logic expressions
    - Optional: simplification
  - Combinational and sequential logic
    - Register transfer notation

- **Computer Architecture**
  - Overview and history
  - The von Neumann machine
  - Datapath, registers, control unit and memory
  - Instruction cycle
  - Clocks, state, sequencing and state machines
  - Layered representations and interpretations

- **Machine Level Data Representation**
  - Bits, bytes and words
  - Numeric data and number bases
  - Fixed- and floating-point numbers
  - Unsigned and signed numbers
  - Non-numeric data
  - Arrays

- **Instruction Set Architecture and Assembly Language**
  - Instruction sets and types
  - Assembly and machine languages

- Instruction formats and addressing modes
- Subroutines
- Memory organization (heap, stack, static, and code segments)

- **I/O and Interrupts**
  - Handshaking, programmed I/O and interrupt-driven I/O
  - Interrupts and interrupt handling structures
  - Memory Maps