COURSE SECTION INFORMATION – EXACT TIMINGS ARE SUBJECT TO COURSE TIMETABLING

Week	Description
One	**Unit 0 – Opening Lecture** **Note: All Week One Lectures are Hybrid because of the suspension of all classes for student orientation on Tuesday, September 8, 2015** **Objectives:** • Provide students with an overview of CST8216 Processor Architecture
One Two Three Four	**Unit 1 – Introductory Electronics Concepts** In this module, the student is introduced to analog circuit design with specific emphasis placed on understanding the concepts directly related to types of circuits found in embedded computing devices. **Objectives:** • Define the following terms: Voltage, Current, Resistance and Power • Characterize the operation of the following devices: Light Emitting Diodes (LEDs), Hex Displays, Resistors, and Switches. • Predict voltage, current, and power values for simple circuits containing combinations of the following devices: Voltage Source, Resistors, a Fuse and Switches. **Lab:** • Build a simple electronic circuit in Multisim™ • Confirm predicted circuit values by the use of test equipment • Troubleshoot electronic circuits using computer-based training aids **Lab:** • Measure/calculate V, I, R and P in simple electronic circuits in Multisim™ • Use capacitors, switches, and displays in simple electronic circuits in Multisim™ • Construct and analyse simple single source, multiple series, parallel and series/parallel resistor electronic circuits in Multisim™ according to a given schematic diagram • Troubleshoot electronic circuits using computer-based training aids **Assignment One (10% of course mark)**: • Use Multism™ to build Electronic Circuits and analyze them using appropriate test equipment

Week	Description
Four	**Unit 2 – Computer Architecture** In this module, the student analyzes circuit diagrams involving embedded computing devices. **Objectives**: • Describe the functionality of a simple Central Processing Unit (CPU)'s components: o Arithmetic Logic Unit (ALU) o Registers o Memory, Input/Output (I/O) Interface o Control Unit o Clock o BUS
Five Six Seven Eight	**Unit 3 – Arithmetic Principles** In this module, the student explores arithmetic principles as they relate to the implementation of embedded computing devices. **Objectives**: • Calculate the results of mathematical and logical operations on signed and unsigned integer binary and integer hexadecimal numbers. • Determine when overflow conditions exist and describe the relationship to processor architecture. • Construct numbers in the following numbering systems: o Hexadecimal and Binary Coded Decimal (BCD) o Two's Complement (signed and unsigned) • Write simple programs in assembly language to be analyzed, debugged, and verified using simulation techniques. • Identify 68HC12 Classes of Instructions, explain their use, and apply them in assembly language programs: o Load and Store o Arithmetic o Compare and Test **Lab:** • Create simple assembly language programs to exercise arithmetic principles • Run simple assembly language programs in the simulator and troubleshoot them **Lab:** • Identify 68HC12 Classes of Instructions, explain their use, and apply them in assembly language programs: Arithmetic, Compare and Test • Develop assembly language programs using Decision Trees, Pseudocode and Logic Instructions. • Predict the results of program execution with the use of a test plan. **Assignment Two (10% of course mark)**: • Use Multism™ to a build Electronic Circuits and write assembly language programs to implement identical functionally and display the results on LEDs and Hex Displays.

Week	Description
Seven	**Unit 4 – Term Test One – Tuesday, October 20, 2015** In this module, the student is assessed on Course Learning Requirements (CLRs) contained in Unit 1 and Unit 2. The term test is worth 10% of the course mark.
Nine Ten Eleven Twelve Thirteen	**Unit 5– Programming Embedded Systems using Assembly Language** In this module the student applies knowledge of programming languages as it relates to embedded computing devices **Objectives**: • Write simple programs in assembly language to be analyzed, debugged, and verified using simulation techniques. • Apply previously listed 68HC12 Classes of Instructions in assembly language programs • Identify 68HC12 Addressing Modes, explain their use, and apply them in assembly language programs: o Inherent o Immediate o Direct and Extended o Relative o Constant Offset Indexed o Constant Indirect o Auto Pre/Post Decrement/Increment Indexed o Accumulator Offset Indexed o Accumulator D Indirect Indexed • Write simple programs in assembly language to be deployed on the Dragon12-Plus(tm) Freescale HCS12 / 9S12 microcontroller trainer. • Practice lab safety with respect to electronic devices that are sensitive to damage caused by Electrostatic Discharge (ESD). **Lab:** • Analyse given problems to develop Flowcharts, write Pseudocode, and create test plans to represent a problem-solution. • Write 68HC12 assembly language programs by adding the following additional Classes of Instructions to the previous inventory of instructions: o Load, Store Move and Transfer o Boolean Logic and Bit Oriented o Shift and Rotate o Branch and Jump o Stack operations • Develop assembly language programs to implement a problem-solution. • Document program development using tools such as Microsoft Visio and Microsoft Word. **Assignment Three (10% of course mark)**: • Write assembly language programs to implement various Classes of Instructions and Addressing Modes and display the results on LEDs, Hex Displays and/or LCDs.
Twelve	**Unit 6 – Term Test Two – Tuesday, November 17, 2015** In this module, the student is assessed on Course Learning Requirements (CLRs) contained in Unit 3 and Unit 5. The term test is worth 10% of the course mark.

Thirteen Fourteen	**Unit 7 – Introductory Digital Concepts** In this module, the student studies the logical operation, application and troubleshooting of logic gates and is is exposed to several types of combinational logic circuits. **Objectives:** • Characterize the operation of Logic Functions and Gates. • Choose appropriate logic functions to solve simple design problems • Determine when a logic gate will pass a digital waveform and when it will block the signal • Construct timing diagrams showing the proper time relationships of inputs and outputs for the various ideal logic gates **Assignment Four (10% of course mark)**: • Problem solve a user-defined scenario using pseudocode and flowcharts and implement a solution in Assembly Language. Structured programming, the creation and use of library subroutines, documentation and the correlation between code and flowcharts will be emphasized. • Write simple programs in the Assembly Language to be deployed on the Dragon12-Plus(tm) Freescale HCS12 / 9S12 microcontroller trainer.
Fifteen	**Unit 8 – Final Examination (during Final Assessment Week) (30% of course mark)** **Objectives:** • To conduct a comprehensive examination on theoretical and practical aspects of course material.