1. **Course Number and Name**
   CPE 211 Introduction to Computer Programming for Engineers

2. **Credits and Contact Hours**
   3 Credits
   Lecture: 40 contact hours per semester
   Lab: 120 minutes of contact hours per week

3. **Instructor/Course Coordinator**
   - **Instructor:** Mr. Ron Bowman
   - **Course Coordinator:** Mr. Ron Bowman

4. **Textbook(s) (title, author, publisher, year)**

5. **Specific Course Information**
   a. **Catalog Description**
      Solution of engineering problems using a digital computer. Hardware structure of the stored-program computer; programming in a high level language such as C or C++, engineering approximation of dynamic systems; top-down design and algorithms. Practice in solving engineering problems.

   b. **Prerequisites or Co-requisites**
      MA 113 or MA115 - Precalculus.
      Students enrolling in CPE 112 must enroll concurrently in CPE112L.

   c. **Required, Elective or Selected Elective Course**
      This course is required

6. **Specific Course Goals**
   a. **Outcomes of Instruction**
      1. Students will utilize various combinations of algebra, trigonometry, and basic statistics in the completion of laboratory programming projects.
      2. Students will demonstrate problem-solving skills when completing laboratory programming assignments
      3. Students will learn how to formulate computer program solutions for problems presented as programming assignments

   b. **Criterion 3 Outcome Addressed by this Course**
      In this course the student will have to show:
      a) An ability to apply knowledge of mathematics, science and engineering.
      e) An ability to identify, formulate and solve engineering problems.

7. **List of Topics Covered**
   - Overview of Programming and Problem Solving
   - C++ Syntax and Semantics, and the Program Development Process
   - Numeric Types, Expressions, and Output
   - Program Input and the Software Design Process
   - Conditions, Logical Expressions, and Selection Control Structures
   - Looping
- Functions
- Scope, Lifetime, and Interface Design
- The Switch, Do-While, For, Break, and Continue Statements
- Simple Data Types: Built-In and User-Defined
- Structs, Data Abstraction
- Arrays
1. **Course Number and Name**  
   CPE 212 Computer Engineering Design I

2. **Credits and Contact Hours**  
   3 Credits  
   Lecture: 40 contact hours per semester

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. Jeffrey Kulick  
   - **Course Coordinator:** Dr. Seong-Moo Yoo  

4. **Textbook(s) (title, author, publisher, year)**  
     a. **Supplemental Materials**  
        - None

5. **Specific Course Information**  
   a. **Catalog Description**  
      Introduction to structured programming using C++. Search and sort algorithms. Introduction to data structures. Applications to engineering related problems.
   
   b. **Prerequisites or Co-requisites**  
      CPE 112 Introduction to Computer Programming for Engineers

   c. **Required, Elective or Selected Elective Course**  
      Required.

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      1. Students will be able to use the techniques, skills, and modern engineering tools necessary for engineering practice.
   
   b. **Criterion 3 Outcome Addressed by this Course**  
      In this course the student will have to show:  
      a) An ability to apply knowledge of mathematics, science, and engineering
7. List of Topics Covered
   - Classes
   - Operator overloading
   - Templates
   - Lists and string ADT
   - Linked lists
   - Stacks and queues
   - Recursion
   - Trees
   - Graphs
   - Searching
   - Hashing
   - Sorting
1. **Course Number and Name**  
   CPE 221 Computer Organization

2. **Credits and Contact Hours**  
   3 Credits  
   Lecture: 40 contact hours per semester

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. Seong-Moo Yoo  
   - Dr. Rhonda Gaede  
   - **Course Coordinator:** Dr. Seong-Moo Yoo

4. **Textbook(s) (title, author, publisher, year)**  
     a. **Supplemental Materials**  
        - None

5. **Specific Course Information**  
   a. **Catalog Description**  
      Functional organization of stored-program digital computers including number representation, assembly language programming, computer hardware, micro-operations, control logic, and microprocessor architecture.

   b. **Prerequisites or Co-requisites**  
      EE 202 Introduction to Digital Logic Design

   c. **Required, Elective or Selected Elective Course**  
      Required.

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      Student will understand the SRC assembly language simulator.

   b. **Criterion 3 Outcome Addressed by this Course**  
      In this course the student will have to show:  
      k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
7. **List of Topics Covered**
   - The General Purpose Machine
   - Machines, Machine Languages, and Digital Logic
   - Processor Design
   - Processor Design – Advanced Topics
   - Computer Arithmetic
   - Memory System Design
   - Input and Output
1. **Course Number and Name**  
CPE 322, Digital Hardware Design Fundamentals

2. **Credits and Contact Hours**  
3 credits hours, 40 contact hours per semester

3. **Instructor/Course Coordinator**  
- **Instructors:** Dr. Rhonda Gaede  
  Dr. Aleksandar Milenkovic  
  Dr. B. Earl Wells  
- **Course Coordinator:** Dr. B. Earl Wells

4. **Textbook(s) (title, author, publisher, year)**  
  a. **Supplemental Materials**  

5. **Specific Course Information**  
   a. **Catalog Description**  
   Advanced concepts in Boolean algebra, use of hardware description languages as a practical means to implement hybrid sequential and combinational designs, digital logic simulation, rapid prototyping techniques, and design for testability concepts. Focuses on the actual design and implementation of sizeable digital design problems using representative Computer Aided Design (CAD) tools
   
   b. **Prerequisites**  
   CPE 221 Computer Organization, Corequisite CPE 324 Advanced Logic Design Laboratory
   
   c. **Required, Elective or Selected Elective Course**  
   Required Course
6. Specific Course Goals
   a. Outcomes of Instruction
      1. Students are required to apply two-value Boolean Algebra techniques to analyze and optimize
         combinational and digital logic circuits.
      2. Homework and simulation problems are presented that allow students the opportunity to create
         digital circuitry that meets the detailed specification.
   b. Criterion 3 Outcome Addressed by this Course
      In this course the student will have to show:
      a) An ability to apply knowledge of mathematics, science and engineering.
      e) An ability to identify, formulate, and solve engineering problems.

7. List of Topics Covered
   • Overview of Digital Logic Fundamentals
   • Timing Issues in Combinational Logic
   • Hazard Avoidance and Detection
   • Sequential Logic Overview
   • Implementation of Moore and Mealy Machines
   • Schematic Capture Implementations
   • Hardware Description Language Implementations
   • Timing issues (setup, hold time constraints, pulse width limitations, maximum
     clock frequency determinations, etc.)
   • Post-synthesis Timing Verification
   • Rapid Prototyping
   • Small-scale Programmable Logic Device Architecture
   • Field Programmable Gate Array and
     Complex Programmable Logic Device Architectures
   • Basic Concepts in Digital Testability
   • Circuit defects and Faults
   • Test pattern generation
   • Fault Coverage, detection and testing issues
1. **Course Number and Name**  
   CPE 323 Introduction to Embedded Computer Systems

2. **Credits and Contact Hours**  
   3 Credits  
   Lecture: 40 contact hours per semester

3. **Instructor/Course Coordinator**  
   • *Instructor(s):* Dr. Aleksandar Milenkovic  
   • *Course Coordinator:* Dr. Aleksandar Milenkovic

4. **Textbook(s) (title, author, publisher, year)**  
   a. *Supplemental Materials*  
      • Lecture slides, Aleksandar Milenkovic, The University of Alabama in Huntsville  
      • TI MSP430 User Guides, TI MSP430 Reference Manuals

5. **Specific Course Information**  
   a. *Catalog Description*  
      The course examines both hardware and software aspects in building embedded computer systems, as well as methods to evaluate design tradeoffs between different technology choices. The students develop an appreciation of technology capabilities and limitations and appreciation of all system components necessary to design and implement a basic embedded computer system and interface it to the outside world. Experiments performed in the Embedded Systems Laboratory provide considerable experience, allowing students to develop programs in assembly language and C and program embedded systems to perform required functions.

   b. *Prerequisites or Co-requisites*  
      Prerequisite: CPE 221 (Computer Organization)  
      Corequisite: CPE 325

   c. *Required, Elective or Selected Elective Course*  
      Required.
6. **Specific Course Goals**
   
   **a. Outcomes of Instruction**
   
   1. Students will analyze and design assembly and C language programs that solve engineering problems and use mathematics to evaluate design metrics (e.g., performance, cost, and energy).
   2. Students will understand hardware/software interfaces and how hardware components can be utilized by software to perform certain tasks in embedded systems.
   
   **b. Criterion 3 Outcome Addressed by this Course**
   In this course the student will have to show:
   a) An ability to apply knowledge of mathematics, science, and engineering.
   b) An understanding of hardware/software co-design issues, especially in the context of real-time, embedded and network systems.

7. **List of Topics Covered**
   
   - Review (computer architecture, number representation)
   - Introduction to Microprocessor-Based System Design
   - TI MSP430 Microcontroller: An Introduction
   - Programmer's View (TI MSP430): Registers, Data types, Memory, Addressing Modes, ISA
   - Software development: Assembly Programming and Debugging
   - Software Development: C/C++
   - TI MSP430 System Architecture
   - Exception Handling, Interrupts, Interrupt Service Routines, Operating modes
   - I/O Interfacing: Parallel Ports
   - Time: Basic clock module, Timers (Watchdog Timer, Timer A, Timer B)
   - Parallel Communication: Simplex, Duplex, Half-duplex, Handshaking
   - Serial Communication: Principles (asynchronous, synchronous), serial communication interfaces and protocols (UART, SPI, I2C)
   - Analog signals: Principles, ADC; DAC peripherals
   - Direct Memory Access Controller: Principles, DMA peripheral
   - LCDs: Principles, LCD controller peripherals
ABET Syllabus CPE 324

1. Course Number and Name
   CPE 324, Advanced Logic Design Laboratory

2. Credits and Contact Hours
   1 credits hour, 3 contact hours per week

3. Instructor/Course Coordinator
   • **Instructors:** Dr. Rhonda Gaede
     Dr. Aleksandar Milenkovic,
     Dr. B. Earl Wells
   • **Course Coordinator:** Dr. B. Earl Wells

4. Textbook(s) (title, author, publisher, year)
     a. Supplemental Materials
        • Handout: Introduction to Report Writing.

5. Specific Course Information
   d. **Catalog Description**
      Demonstration through laboratory experimentation of fundamental concepts in digital logic design. Use of hardware description languages as a practical means to implement hybrid sequential and combinatorial digital designs, digital logic simulation, and rapid prototyping techniques. Focuses on the actual design and implementation of sizeable digital design problems using representative Computer Aided Design (CAD) tools.
   b. **Prerequisites**
      Corequisite: CPE 322 Digital Hardware Design Fundamentals
   c. **Required, Elective or Selected Elective Course**
      Required Course

6. Specific Course Goals
   a. **Outcomes of Instruction**
      In this course student projects
      1. Will be primarily focused around the task of developing a component in reconfigurable hardware that integrates well within an already existing IP Core framework. At least one of these projects will require the development of designs that interact with external physical and electronic peripheral devices. This will require a clear understanding of physical timing constraints as well as established digital protocols.
      2. Will commonly require the skillful use of modern design entry, analysis and synthesis tools. Student projects will occasionally require the use of digital simulation techniques and tools.
      3. Will, in at least one case, allow the student to explore alternative hardware and software based approaches to meeting the design requirements. In this process they will be required to evaluate the trade-offs associated with these different types of representations that span the software/hardware continuum.
   b. **Criterion 3 Outcome Addressed by this Course**
      In this course the student will have to show:
      c) Ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
      k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering.
      4. Students will be able to apply engineering requirements and standards
1) An understanding of hardware/software co-design issues, especially in the context of real-time, embedded and network systems.

7. **List of Topics Covered**
   - Sequential circuit design and implementation. Design and implementation of a simple Mealy or Moore state sequencer using traditional finite state machine and Boolean logic simplification techniques. Implementation of design using SSI components on a digital trainer.
   - Introduction to modern simulation and rapid prototyping design methodologies. -- ModelSim and Altera Quartus II ® Computer Aided Design and Simulation Software.
   - Simulation and implementation using an FPGA-based rapid prototype platform of a moderately sized combinational design. Multiple implementations are made using hierarchical schematic capture as well as structural and behavioral hardware description methods.
   - Implementation of a practical combined sequential and combinational design using hardware description language based design methodologies. Example design problem low-level PC keyboard interface.
   - Combined Analog digital design. Typical design problem -- a Capacitance Tester.
   - Common Computer Hardware Implementation. Typical design problem: Low-level Video controller design.
   - Hardware/Software Trade-off design. Students explore design trade-offs associated with an embedded system type design that can be implemented in hardware or software. Focus will be on hardware implementation but students will be presented alternative software dominate solutions and required to evaluate the trade-offs to both general approaches.
1. **Course Number and Name**  
   CPE 325 Embedded Systems Laboratory

2. **Credits and Contact Hours**  
   1 Credit  
   Contact Hours: Two 80 minute laboratory sessions per week

3. **Instructor/Course Coordinator**  
   - *Instructor:* Dr. Aleksandar Milenkovic  
   - *Course Coordinator:* Dr. Aleksandar Milenkovic

4. **Textbook(s) (title, author, publisher, year)**  
     a. *Supplemental Materials*  
        - Laboratory tutorials.  
        - TI MSP430 User Guides, TI MSP430 Reference Manuals

5. **Specific Course Information**  
   a. **Catalog Description**  
      Laboratory component of Introduction to Embedded Computer Systems. Students will gain considerable experience working with modern integrated software development environments and hardware platforms to solve practical problems. The experiments will include software development in assembly and C/C++; debugging and testing of embedded systems; and input/output interfacing including timers, serial communication interfaces, ports, analog-to-digital converters, digital-to-analog converters, and direct memory-access controllers.

   b. **Prerequisites or Co-requisites**  
      Co-requisite: CPE 323

   c. **Required, Elective or Selected Elective Course**  
      Required.

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      1. Students will learn how to develop, verify, and test software for modern embedded systems to meet desired needs within realistic constraints (functionality, performance, power consumed)

         2. Students will use industry leading software development environments for embedded systems and hardware development platforms.

         3. Students will explore hardware-supported and software-only implementations of particular problems.

   b. **Criterion 3 Outcome Addressed by this Course**  
      In this course the student will have to show:  
      c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability  
      k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
1) An understanding of hardware/software co-design issues, especially in the context of real-time, embedded and network systems.

7. List of Topics Covered
   - Integrated Development Environments (IDE) for Embedded Systems
   - MSP430-based Hardware Development Platform (PCB, Schematics, Interfacing)
   - Software development for embedded systems in assembly language
   - Software development for embedded systems in C/C++
   - Debugging, program download, testing
   - Time in embedded systems: clocks, timers, watchdog timers
   - Ports: parallel ports, LEDs, display controllers, switches
   - Serial communication interfaces (UART, SPI, I2C)
   - Analog peripherals: comparators, ADC, DAC
1. **Course Number and Name**  
   CPE 353 – Software Design and Engineering

2. **Credits and Contact Hours**  
   3 credits, two 80 minutes lectures per week

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. David Coe  
   - **Course Coordinator:** David J. Coe

4. **Textbook(s) (title, author, publisher, year)**  
   a. **Supplemental Materials**  
      - Lecture Notes, David J. Coe, University of Alabama in Huntsville  
      - Online Qt documentation, examples, and tutorials: [http://qt-project.org/doc/](http://qt-project.org/doc/)

5. **Specific Course Information**  
   a. **Catalog Description**  
      Hands-on experience developing a substantial software project using software design tools such as the SQL database system and the Qt graphical interface development environment. Introduction to a software process including requirements elicitation and testing techniques
   b. **Prerequisites or Co-requisites**  
      Prerequisite: CPE 212. Prerequisite with concurrency: CS 317
   c. **Required, Elective or Selected Elective Course**  
      Required

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      c) Provides an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
   b. **Criterion 3 Outcome Addressed by this Course**  
      In this course the student will have to show:  
      c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

7. **List of Topics Covered**  
   - Qt4 Graphical User Interface (GUI) Framework  
   - Basic Qt4 Widgets and Their Integration into Applications  
   - Networked Qt Applications  
   - Qt Threads and Synchronization Primitives  
   - Unit Testing within Qt4  
   - Introduction to Relational Algebra and SQL programming
- Integration of SQL databases with Qt Applications
- User Interface Design Principles
- User Interface Development Process
- Usability Evaluation Techniques
- User Interface Testing
1. **Course Number and Name**  
   CPE 381, Fundamentals of Signals and Systems for CPE

2. **Credits and Contact Hours**  
   3 credit hours, 40 contact hours per semester

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. Emil Jovanov  
   - **Course Coordinator:** Dr. David Pan

4. **Textbook(s) (title, author, publisher, year)**  
   - *Supplemental Materials*  
     - Instructor Course Handout

5. **Specific Course Information**  
   a. **Catalog Description**  
      Introduction to the fundamental concepts in continuous and discrete signals and systems, and methods of signal and system analysis. Topics covered: Fourier series, Fourier and Laplace transforms, system representation by transfer functions and impulse response functions, convolution integrals, discrete time signals and systems. Sampling techniques, Z and discrete Fourier transforms.

   b. **Prerequisites or Co-requisites**  
      EE 213 and MA 238

   c. **Required, Elective or Selected Elective Course**  
      Required Course

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      Students will understand the fundamental concepts in continuous and discrete signals and systems and use various transform-based methods for signal and system analysis.

   b. **Criterion 3 Outcome Addressed by this Course**  
      a) Students will apply mathematics, science and engineering.

7. **List of Topics Covered**  
   - Introduction to Signals and Systems  
   - Linear Time – Invariant Systems  
   - Fourier Series  
   - Fourier Transforms  
   - Discrete Time Fourier Series  
   - Discrete Time Fourier Transforms  
   - Sampling  
   - Laplace Transform  
   - Z Transform
1. **Course Number and Name**  
   CPE 412, Introduction to Parallel Programming

2. **Credits and Contact Hours**  
   3 credits hours, 40 contact hours per semester

3. **Instructor/Course Coordinator**  
   - **Instructors:** Dr. B. Earl Wells  
   - **Course Coordinator:** Dr. B. Earl Wells

4. **Textbook(s) (title, author, publisher, year)**  
     a. **Supplemental Materials**  

5. **Specific Course Information**  
   a. **Catalog Description**  
      Introduction to software development methodologies for multi-core, clustered, and general purpose GPU computing environments. Exploration of parallel machine models, performance evaluation, process and thread creation, mutual exclusion, synchronization, and message passing communication mechanisms. Students gain insight into these concepts through hands-on exercises written in C/C++ that employ the MPI, OpenMP, and OpenACC SDKs.

   b. **Prerequisites**  
      CPE 212 Fundamentals of Software Engineering  
      CS 317 Introduction to Design and Analysis of Algorithms

   c. **Required, Elective or Selected Elective Course**  
      Elective Course
6. **Specific Course Goals**
   
   **a. Outcomes of Instruction**
   
   1. Students in this course are required to develop parallel representations that meet specified performance constraints. Meeting these performance constraints require that students make the necessary trade-off analysis associated with the gains in performance that result from parallelization as contrasted with the performance degradation that results from increased communication and synchronization present when the computation is spread out among a larger number of processing elements.
   
   2. Students in this course are required to identify an application problem that can benefit from parallel or distributed processing and to formulate a solution of this problem that is then implemented on one or more multi-core, multi-threaded, clustered computing environment, or GPU computing environment.
   
   3. Students are required to utilize modern parallel shared-memory and message-passing tools and libraries such as MPI, OpenMP, and OpenACC. They are also expected to be able to understand and utilize output from parallel profilers to determine performance bottlenecks in their code.

   **b. Criterion 3 Outcome Addressed by this Course**
   
   In this course the student will have to show:
   
   c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
   
   e) an ability to identify, formulate, and solve engineering problems
   
   k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Students will be able to prepare deliverables for an engineering design project.

7. **List of Topics Covered**

   - architectural structure and program model of parallel computers
   - message-passing programming with MPI
   - shared memory programming OpenMP
   - embarrassingly parallel computations
   - partitioning and divide-and-conquer strategies
   - pipelined computations
   - synchronous computations
   - load balancing (static and dynamic)
   - programming with shared memory (thread models/synchronization methods)
   - GPU architectures – comparisons with CPUs
   - Multithreaded computation on GPU’s utilizing OpenACC
1. **Course Number and Name**  
   CPE 423 Hardware/Software Co-design

2. **Credits and Contact Hours**  
   3 Credits  
   Lecture: Two 80 minute lectures per week

3. **Instructor/Course Coordinator**  
   • **Instructor:** Dr. Aleksandar Milenkovic  
   • Dr. Jeffrey Kulick  
   • **Course Coordinator:** Dr. Aleksandar Milenkovic

4. **Textbook(s) (title, author, publisher, year)**  
   • Embedded SoPC Design With Nios II Processor and Verilog Examples, Pong P. Chu, John Wiley and Sons 2012.  
   a. **Supplemental Materials**  
      • Lecture slides, Nios2 Tutorials

5. **Specific Course Information**  
   a. **Catalog Description**  
      This course focuses on electronic systems whose functionality is realized partially in hardware and partially in software. This includes a study of modeling techniques of hardware and software components at different levels of abstraction and a study of interfacing techniques between hardware components and software components. The course includes homework and practical experiments. The students having completed this course, you will have experience in the following: (a) transforming simple software programs into cycle-based hardware descriptions with equivalent behavior and vice versa; (b) partitioning simple software programs into hardware and software components, and create appropriate hardware-software interfaces to reflect this partitioning; (c) Analyzing and explaining the control-flow and data-flow of a software program and a cycle-based hardware description; (d) identifying performance bottlenecks in a given hw/sw architecture and optimize them by transformations on hardware and software components.

   b. **Prerequisites or Co-requisites**  
      CPE 322 or permission of instructor.

   c. **Required, Elective or Selected Elective Course**  
      Elective

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      Students will design an embedded computer system, both hardware and software, to meet desired needs within realistic constraints (functionality, performance, power consumed)

   b. **Criterion 3 Outcome Addressed by this Course**  
      In this course the student will have to show:  
      1) an understanding of hardware/software co-design issues, especially in the context of real-time, embedded and network systems

7. **List of Topics Covered**  
   • Review (computer architecture, embedded systems)
• HDL constructs and digital synthesis
  (combinational, sequential, FSM, FSMD, RTL)
• Design metrics: performance, power, on-chip area
• Programmable chips (FPGAs)
• Soft-core processors: Nios II
• Embedded software
• IP cores design and integration (I/O interface, Avalon interface, SDRAM)
• Hardware accelerators
1. **Course Number and Name**  
   CPE 426 Introduction to VLSI Design Using HDLs, Modeling and Synthesis

2. **Credits and Contact Hours**  
   3 credit hours, 40 contact hours per semester

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. Rhonda Gaede  
   - **Course Coordinator:** Dr. Rhonda Gaede

4. **Textbook(s) (title, author, publisher, year)**  
     
     **a. Supplemental Materials**  

5. **Specific Course Information**  
   a. **Catalog Description**  
      Modern VLSI design techniques and tools, such as silicon compilers, (V)HDL modeling languages, placement and routing tools, synthesis tools, and simulators. Students will design, simulate, and layout using both programmable logic families and ASIC libraries.

   b. **Prerequisites or Co-requisites**  
      EE 202 Introduction to Digital Logic Design  
      EE 315 Introduction to Electronic Analysis and Design

   c. **Required, Elective or Selected Elective Course**  
      Elective

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      1. To understand top-down and behavioral design.  
      2. To generate synthesizable behavioral VHDL models.  
      3. To understand standard cell and FPGA design.  
      4. To develop an appreciation for synthesis algorithms.  
      5. To develop transactional verification of designs using System Verilog
a. **Criterion 3 Outcome Addressed by this Course**
   In this course the student will have to show:
   
c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
   
k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Students will be able to prepare deliverables for an engineering design project.

7. **List of Topics Covered**
   - Modeling Digital Systems, Domains and Levels of Modeling, Modeling Languages, VHDL Modeling Concepts
   - Constants and Variables, Scalar Types, Type Classification, Attributes of Scalar Types, Expressions and Predefined Operations
   - If Statements, Case Statements, Null Statements, Loop Statements, Assertion and Report Statements
   - Arrays, Unconstrained Array Types, Array Operations and Referencing, Records
   - Entity Declarations and Architecture Bodies, Behavioral Descriptions, Structural Descriptions, Design Processing
   - Procedures, Procedure Parameters, Concurrent Procedure Call Statements, Functions, Overloading, Visibility of Declaration
   - Package Declarations, Package Bodies, Use Clauses
   - The Predefined Packages standard and env, IEEE Standard Packages
   - Generic Constants, Generic Types, Generic Lists in Packages, Generic Lists in Subprograms, Generic Packages
   - Components, Configuring Component Instances, Configuration Specifications
   - Generating Iterative Structures, Conditionally Generating Structures, Configuration of Generate Statements
   - Design for Synthesis
   - Functional Verification Using SystemVerilog
a. Course Number and Name
   CPE 427 VLSI Design I

b. Credits and Contact Hours
   3 Credits
   Lecture: 40 contact hours per semester
   Lab: 80 minutes of contact time per week

c. Instructor/Course Coordinator
   • Instructor(s): Dr. Aleksandar Milenkovic
   • Course Coordinator: Dr. Aleksandar Milenkovic


d. Textbook(s) (title, author, publisher, year)
   • Neil H.E. Weste, David Harris,
     CMOS VLSI Design: A Circuits and System Perspective, Addison Wesley,

   a. Supplemental Materials
      • Tool tutorials

e. Specific Course Information
   a. Catalog Description
      The course gives an introduction to digital integrated circuits.
      It covers the following topics. CMOS devices and manufacturing technology. CMOS inverter and
      gates. Propagation delay, noise margins, and power dissipation. Sequential circuits, arithmetic,
      interconnect, and memories. Design methodologies. A major part of the course will be a design
      project.

   b. Prerequisites or Co-requisites
      EE 202 Introduction to Digital Logic Design,
      EE 315 Introduction to Electronic Analysis and Design

   c. Required, Elective or Selected Elective Course
      Elective.
f. **Specific Course Goals**
   
a. **Outcomes of Instruction**
   
   1. Students will apply knowledge in mathematics to design and analyze hardware modules.
   2. Students will use industry leading electronic design automation tools for digital integrated circuits.

   b. **Criterion 3 Outcome Addressed by this Course**
   
   In this course the student will have to show:
   a) An ability to apply knowledge of mathematics, science, and engineering.
   k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

   g. **List of Topics Covered**
   
   - Introduction into Digital Integrated Circuits, Manufacturing
   - The devices: transistors, resistors, parasitic capacitance
   - CMOS Inverter
   - CMOS Logic, Pass Logic
   - Dynamic view
   - Design for speed, power
   - Sequential Logic Cells, Sequential Logic and Gating
   - Alternative design styles (dynamic, domino, pseudo nMOS)
   - Datapath elements (Adders & Multipliers, Shifters)
   - Class Presentations
1. **Course Number and Name**  
   CPE 431 Introduction to Computer Architecture

2. **Credits and Contact Hours**  
   3 hours, 3 contact hours per week

3. **Instructor/Course Coordinator**  
   • **Instructor:** Dr. Rhonda Gaede  
   • **Course Coordinator:** Dr. Rhonda Gaede

4. **Textbook(s) (title, author, publisher, year)**  
     a. **Supplemental Materials**  
        • CPE 431 Instructor Course Handout

5. **Specific Course Information**  
   b. **Catalog Description**  
      Study of existing computer structures. Computer organization with emphasis on busing systems, storage systems, and instruction sets. Performance models and measures, pipelining, cache and virtual memory, introduction to parallel processing.

   c. **Prerequisites or Co-requisites**  
      CPE 221 Computer Organization

   d. **Required, Elective or Selected Elective Course**  
      Required Course

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      1. Students will understand the complex nature of computer performance.  
      2. Students will be able to implement a pipelined reduced instruction set computer (RISC) architecture  
      3. Students will understand advanced architectural issues such as caching and I/O.  
      4. Students will understand the basics of distributed processing.
b. Criterion 3 Outcome Addressed by this Course
   In this course, the student will have to show:
   g) an ability to communicate effectively
   i) a recognition of the need for, and an ability to engage in life-long learning

7. List of Topics Covered
   • Computer Abstraction and Technology
   • Performance
   • Instructions: Language of the Computer
   • The Processor: Datapath and Control
   • Large and Fast: Exploiting Memory Hierarchy
   • Storage and Other I/O Topics
   • Multicores, Multiprocessor, and Clusters
1. **Course Number and Name**  
   CPE 434/534 Operating System

2. **Credits and Contact Hours**  
   3 Credits, 2 x 1 hour 20 minute lectures

3. **Instructor/Course Coordinator**  
   • **Instructor:** Dr. Jeffrey. Kulick,  
   • **Course Coordinator:** J. Kulick,

4. **Textbook(s) (title, author, publisher, year)**  
   
   a. **Supplemental Materials**  
      • none

5. **Specific Course Information**  
   a. **Catalog Description**  
      Study of the fundamentals of Operating Systems. Emphasis on processes, file management, inter-process communication, input output, virtual memory, networking and security.
   b. **Prerequisites or Co-requisites**  
      Co-Requisite CPE 435 Operating Systems Lab
   c. **Required, Elective or Selected Elective Course**  
      Elective

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      1. Students will explore the concepts associated with multi-process and multi-threaded operating systems including understanding performance metrics  
      2. Students will also gain experience writing a research/review type paper that explores modern concepts in operating systems.
   b. **Criterion 3 Outcome Addressed by this Course**  
      In this course, the student will have to show:  
      e) An ability to identify, formulate, and solve engineering problems  
      g) an ability to communicate effectively
7. **List of Topics Covered**

Chapter 1: Introduction to Operating Systems
Chapter 2: Computer System Structures
Chapter 3: Operating Systems Structure

Part II Process Management
Chapter 4: Processes
Chapter 5: Threads
Chapter 6: CPU Scheduling
Chapter 7: Process Synchronization

Part III Storage Management
Chapter 9: Memory Management
Chapter 10: Virtual Memory
Chapter 11: File system Interface
Chapter 12: File System Implementation

Part IV I/O Systems
Chapter 13: I/O Systems
Chapter 14: Mass-Storage Structures

Part V Distributed Systems
Chapter 15: Distributed System Structures
  Network Types
  Communication Protocols
  Name services
1. **Course Number and Name**  
   CPE 435 Operating Systems Laboratory

2. **Credits and Contact Hours**  
   1 Credit  
   Contact Hours: Two 80 minute laboratory sessions per week

3. **Instructor/Course Coordinator**  
   • *Instructor:* J. Kulick, Professor, M. Milenkovic, Instructor, ECE Dept.  
   • *Course Coordinator:* J. Kulick, Professor, ECE Dept

4. **Textbook(s) (title, author, publisher, year)**  
   
   - **Supplemental Materials**  
     • none

5. **Specific Course Information**  
   d. *Catalog Description*  
      Laboratory component of operating systems course. Experiments include implementation of device drivers, process and thread management, virtual memory management, dynamic memory management, file systems.

   e. *Prerequisites or Co-requisites*  

   f. *Required, Elective or Selected Elective Course*  
      Required

6. **Specific Course Goals**  
   a. *Outcomes of Instruction*  
      1. Students shall be able to make performance measurements on their programs using contemporary performance analysis tools.
      2. Students shall implement an application in multiple ways using different technologies.

   b. *Criterion 3 Outcome Addressed by this Course*  
      In this course, the student will have to show:
      b) an ability to design and conduct experiments, as well as to analyze and interpret data  
      c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

7. **List of Topics Covered**  
   Chapter 1: Introduction to Operating Systems  
   Chapter 2: Computer System Structures  
   Chapter 3: Operating Systems Structure

   Part II Process Management  
   Chapter 4: Processes  
   Chapter 5: Threads  
   Chapter 6: CPU Scheduling  
   Chapter 7: Process Synchronization

   Part III Storage Management
1. **Course Number and Name**  
   CPE 436/536 Internals of a Modern Operating System

2. **Credits and Contact Hours**  
   3 Credits, 2 x 1 hour 20 minute lectures

3. **Instructor/Course Coordinator**  
   • **Instructor:** J. Kulick, Professor, M. Milenkovic, Instructor, ECE Dept.  
   • **Course Coordinator:** J. Kulick, Professor, ECE Dept

4. **Textbook(s) (title, author, publisher, year)**  
   • Linux Kernel Architecture, W. Maurer, WROX press, 2014  
   • Understanding the Linux Kernel, D. Bovet, M Cesati, O’Reilley, 2005  
   c. **Supplemental Materials**  
      • none

5. **Specific Course Information**  
   g. **Catalog Description**  
      In-depth study of the design of modern operating systems such as Unix, NT and Linux. Emphasis on the internals and implementation details of interrupt processing, real-time clocks, device independent I/O, process management, memory management, file management.

   h. **Prerequisites or Co-requisites**  
      Prerequisite: CPE 434 Introduction to Operating Systems

   i. **Required, Elective or Selected Elective Course**  
      Elective

6. **Specific Course Goals**  
   a. **Outcomes of Instruction**  
      to understand how modern operating systems are constructed,  
      to understand issues with security and safety related to operating system design,  
      to participate in lifelong learning and communications
b. **Criterion 3 Outcome Addressed by this Course**
In this course, the student will have to show:
  c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
  g) an ability to communicate effectively
  i) a recognition of the need for, and an ability to engage in life-long learning

7. **List of Topics Covered**
   - Process Management and Scheduling
   - Memory Management
   - Virtual Process Memory
   - Locking and Interprocess Communication
   - Device Drivers
   - Modules
   - The Virtual File System
   - Extended File Systems
   - Security and Access Control Lists
   - Networks
   - Kernel Time Management
   - Paging and Synchronization
1. **Course Number and Name**
   CPE 448, Introduction to Computer Networks

2. **Credits and Contact Hours**
   3 credit hours, 40 contact hours per semester

3. **Instructor/Course Coordinator**
   - **Instructor:** Mr. Ron Bowman, Instructor, Electrical and Computer Engineering
   - **Course Coordinator:** Dr. David Pan, Associate Professor, Electrical and Computer Engineering

4. **Textbook(s) (title, author, publisher, year)**
     a. **Supplemental Materials**
        - Instructor Course Handout

5. **Specific Course Information**
   a. **Catalog Description**
      Introduction to the basic computer network concepts, underlying technologies, including Local Area Networks, Ethernet, Internet, TCP/IP and Application Layer Protocols, Socket Programming, Data Compression, and Network Security.
   b. **Prerequisites or Co-requisites**
      CPE 112 and CPE 221
   c. **Required, Elective or Selected Elective Course**
      Required Course

6. **Specific Course Goals**
   a. **Outcomes of Instruction**
      Students will understand the fundamental computer network concepts and the TCP/IP protocols and will be able to program with network APIs.
   b. **Criterion 3 Outcome Addressed by this Course**
      In this course, the student will have to show:
      k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
7. List of Topics Covered
   • Topologies of Computer Networks
   • Layered Network Protocols Architectures
   • Physical Interfaces
   • Multiplexing Techniques
   • Data Link Control Protocols
   • Switching Techniques
   • Medium Access Control Methods
   • Network Layer Routing Protocols
   • Transport Layer Protocols
   • Application Layer Protocols
   • Data Compression
   • Network Security
1. **Course Number and Name**  
   CPE 453 – Senior Software Studio

2. **Credits and Contact Hours**  
   3 credits, two 80 minutes lectures per week

3. **Instructor/Course Coordinator**  
   - **Instructor:** J. Kulick, Professor, ECE Dept  
   - **Course Coordinator:** David J. Coe, Associate Professor, ECE Department

4. **Textbook(s) (title, author, publisher, year)**  
   - **Supplemental Materials**  
     - Lecture Notes, David J. Coe, University of Alabama in Huntsville  
     - Online Qt documentation, examples, and tutorials: [http://qt-project.org/doc/](http://qt-project.org/doc/)

5. **Specific Course Information**  
   - **Catalog Description**  
     Basic concepts of software engineering. Software project management including specification, design, implementation, testing, and documentation. Software design and management tools. Includes a multi-student software project.
   
   - **Prerequisites or Co-requisites**  
     CS 317 and CPE 353
   
   - **Required, Elective or Selected Elective Course**  
     Required
6. **Specific Course Goals**
   
   **a. Outcomes of Instruction**
   Students work in small teams (2-4 students) to develop a non-trivial software product. Students learn basic project management skills including estimation, planning, tracking, and progress reporting.
   Students learn to work as a team
   Students identify and scope a project to meet course objectives
   Students propose software projects that satisfy course project constraints.
   Students formulate solution requirements and constraints.
   Students implement and test their software solution for the problem.
   Students use modern engineering tools when developing their software products
   Students use modern, cross-platform graphical interface development tool
   Students use industry standard database tools

   **b. Criterion 3 Outcome Addressed by this Course**
   In this course, the student will have to show:
   d) an ability to function on multi-disciplinary teams
   e) an ability to identify, formulate, and solve engineering problems
   k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

7. **List of Topics Covered**
   - Agile Software Development
   - Test-Driven Development
   - Team Software Development
   - Software Testing Metrics
   - Configuration Management
   - Software Project Estimation, Planning, and Tracking
1. **Course Number and Name**  
   CPE 495 Computer Engineering Design I

2. **Credits and Contact Hours**  
   3 Credits  
   Lecture: Two 80 minute lectures per week

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. B. Earl Wells, Professor, Electrical and Computer Engineering  
   - **Course Coordinator:** Dr. Emil Jovanov, Associate Professor, Electrical and Computer Engineering

4. **Textbook(s) (title, author, publisher, year)**  
   a. **Supplemental Materials**  
      - None

5. **Specific Course Information**  
   a. **Catalog Description**  
      This is the first in the two-course senior capstone design sequence. The focus of this class is the application of techniques to the design of electronic systems that have digital hardware and software components. Students will apply the knowledge and skills acquired in earlier course work and incorporate appropriate engineering standards to solve real-world design problems and design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

   b. **Prerequisites or Co-requisites**  
      CPE353 (Software Design and Engineering), EE 315 (Introduction to Electronic Analysis and Design), and CPE323 (Introduction to Embedded Systems). Must be taken in the same academic year as CPE 496.

   c. **Required, Elective or Selected Elective Course**  
      Required.
6. Specific Course Goals
   
   a. Outcomes of Instruction
      
The major goal of this course is to provide a vehicle through which students can apply the knowledge and skills acquired in earlier course work and incorporate appropriate engineering standards to solve real-world design problems and design a system, component, or process to meet a customer’s needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. Their solution space should involve the careful consideration of hardware/software design trade-offs. Students are to perform their work as part of a larger team and must be able to successfully communicate their results to the engineering community both in written and oral format. The scope of their design experience should be technically challenging while providing these CPE students with a venue to demonstrate professionalism and underscore the need for ethical responsibility.

   b. Criterion 3 Outcome Addressed by this Course
      
      In this course, the student will have to show:
      
      d) an ability to function on multi-disciplinary teams
      f) an understanding of professional and ethical responsibility
      g) an ability to communicate effectively
      h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and social context
      k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

7. List of Topics Covered
   
   • Introduction, Teams and Teamwork, Projects
   • Rapid Prototyping Techniques and PCB Design Process
   • The Engineering Design Process, Hardware/software trade-offs for digital system design
   • Low Power Embedded Microcontrollers: Design Issues
   • Design Process presentation and test
   • Testing and System Reliability
   • Concept Generation and Evaluation,
   • System Design, Project Management, Oral Presentations Guidelines
   • Ethical and Legal Issues
1. **Course Number and Name**  
   CPE 496 Computer Engineering Design II

2. **Credits and Contact Hours**  
   3 Credits  
   Lecture: Two 80 minute lectures per week

3. **Instructor/Course Coordinator**  
   - **Instructor:** Dr. B. Earl Wells, Professor, Electrical and Computer Engineering  
   - **Course Coordinator:** Dr. Emil Jovanov, Associate Professor, Electrical and Computer Engineering

4. **Textbook(s) (title, author, publisher, year)**  
     a. **Supplemental Materials**  
        - None

5. **Specific Course Information**  
   a. **Catalog Description**  
      Second course in the senior capstone design sequence. The focus of this class is the application of techniques to the design of electronic systems that have digital hardware and software components. Students will apply the theory acquired from numerous engineering courses to solve real-world design problems. Prerequisite: CPE 495. Must be taken in the same academic year as CPE 495.

   b. **Prerequisites or Co-requisites**  
      CPE 495 Computer Engineering Design I

   c. **Required, Elective or Selected Elective Course**  
      Required.
6. Specific Course Goals
   a. Outcomes of Instruction
      The major goal of this course is to provide a vehicle through which students can apply the knowledge and skills acquired in earlier course work and incorporate appropriate engineering standards to solve real-world design problems and design a system, component, or process to meet a customer’s needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. Their solution space should involve the careful consideration of hardware/software design trade-offs. Students are to perform their work as part of a larger team and must be able to successfully communicate their results to the engineering community both in written and oral format. The scope of their design experience should be technically challenging while providing these CPE students with a venue to demonstrate professionalism and underscore the need for ethical responsibility.

   b. Criterion 3 Outcome Addressed by this Course
      In this course, the student will have to show:
      d) an ability to function on multi-disciplinary teams
      e) an ability to identify, formulate, and solve engineering problems
      g) an ability to communicate effectively
      i) a recognition of the need for, and an ability to engage in life-long learning
      j) a knowledge of contemporary issues

7. List of Topics Covered
   • Project Management Issues
   • Design Presentations
   • Hardware/Software Design Trade-off Justifications