COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 2102</th>
<th>Course Title</th>
<th>Discrete Mathematics I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Knight</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td>Date of Preparation</td>
<td>6/1/2009</td>
</tr>
</tbody>
</table>

Current Catalog Description

Introduces discrete mathematics and proof techniques involving first order predicate logic and induction. Application areas include sets (finite and infinite), elementary combinatorial problems, and probability. Development of tools and mechanisms for reasoning about discrete problems. Cross-listed as APMA 202.

Prerequisite: CS 1110 (101) or 1120 (150) with a grade of C- or higher.

Textbook


References

- Other readings on the class Web site.

Course Outcomes

Upon completion of this course students will have the ability to:

- Introduce a formal system (propositional and predicate logic) which mathematical reasoning is based on.
- Develop an understanding of how to read and construct valid mathematical arguments (proofs) and understand mathematical statements theorems.
- Introduce and work with various problem solving strategies and techniques (e.g. abstracting away unimportant details to state a problem in its basic form, transforming a problem into another simpler problem which we have already solved).
- Introduce and work with important discrete data structures such as sets, relations, and discrete functions.
- Develop important mathematical traits such as precision and the willingness and ability to investigate and solve one problem in multiple ways to verify that the solution is indeed correct.
Class/Laboratory schedule

- This course does not have schedule labs.
- Elements of grade:
  - Ten weekly assignments 30%
  - Semester examination 1 20%
  - Semester examination 2 20%
  - Final examination 30%
- Topics covered in class in this order:
  - Set theory (core concepts)
  - Propositions (core concepts)
  - Predicates (core concepts)
  - Overview of formal specifications
  - Declarative Languages
  - Propositions (additional concepts)
  - Predicates (additional concepts)
  - Proof techniques
  - Proofs for software
  - Graph Theory
  - Permutations and combinations
  - State Machines and Grammars

Topics Covered

propositional logic
set theory
predicate calculus
relations and functions
basic proof techniques
combinatorics and algorithm analysis
elementary probability
declarative formal languages
graph theory
finite state automata
application of these techniques to software specification, verification and analysis

Prerequisites by Topic

- A first course in programming (e.g. CS1110 or any CS1 course)
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 2150</th>
<th><strong>Course Title</strong></th>
<th>Program and Data Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Bloomfield</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td>Date of Preparation</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

This course will introduce you to program and data representation from high level concepts to machine level implementation. Data structuring techniques and the representation of data structures during program execution are discussed, as are operations and control structures and their representation during program execution. Representations of numbers, arithmetic operations, arrays, records, recursion, hashing, stacks, queues, trees, graphs, and related concepts are also covered.

Textbook


Course Outcomes

*Understand program representation from the high-level programming language perspective down to the underlying machine level representation, including: number representation, operations, conditionals, and control structures
*Be able to implement basic and advanced abstract data types in C++ including: linked lists, stacks, queues, hash tables, trees, and graphs
*Be able to evaluate asymptotic time and space complexity analysis of programs and data structure implementations using Big-O, Big-Omega, and Big-Theta notation and assess the suitability of a data structure for a particular problem
*Understand the basic program execution model and the underlying computer hardware and software (fetch-execute cycle, memory hierarchy, operating system, compiler)
*Be able to implement basic program control and data structures in an assembly language (loops, conditionals, subroutines and parameter passing modes, arrays)
Class/Laboratory schedule

<table>
<thead>
<tr>
<th>Laboratory Meetings (all in Olsson 001):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 1: Tuesday, 3:30 – 5:15 p.m.</td>
</tr>
<tr>
<td>Section 2: Tuesday, 5:30 – 7:15 p.m.</td>
</tr>
<tr>
<td>Section 3: Tuesday, 7:30 – 9:15 p.m.</td>
</tr>
<tr>
<td>There will be 12 programming labs this semester</td>
</tr>
</tbody>
</table>

Prerequisites by Topic

| CS 2110 (aka 201) (Software Development Methods) or CS 2220 (aka 205) (Engineering Software); and CS 2102 (aka 202) (Discrete Math), all with grades of C- or above. Note that CS 2102 (aka 202) is allowed as a co-requisite, but see the details for this in the first slide set. |
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 3205</th>
<th>Course Title</th>
<th>HCI in Software Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Horton</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Introduces programs and data representation at the machine level. Data structuring techniques and the representation of data structures during program execution. Operations and control structures and their representation during program execution. Representations of numbers, arithmetic operations, arrays, records, recursion, hashing, stacks, queues, trees, graphs, and related concepts. Prerequisite: CS 202 and either CS 201 or CS 205 with all grades of C- or higher.

Textbook


References

*Task-Centered User Interface Design,* on the Web at http://hcibib.org/tcuid
Other readings on the class Web site.

Course Outcomes

Upon completion of this course students will have the ability to:

- Comprehend fundamental principles of HCI and user-centered design. Evaluate software user interfaces based on defined usability criteria, using methods such as heuristic evaluation and user observation techniques. Apply user-centered design and usability engineering principles as they design a variety of software user interfaces. Use prototyping methods to discover requirements and to evaluate design alternatives. Conduct simple formal experiments to evaluate usability hypotheses. Use GUI development libraries and tools to create usable interfaces for simple windowed software applications.
Class/Laboratory schedule

Team Project (30% total grade, group assignment):
Teams of four students will work on a three-part project (each worth 10%).
In Part 1, you will define and evaluate a task or problem in the real world, and
define a set of requirements for a software system to address this.
In Part 2, you will develop three low-fidelity prototypes for part of this system, and
evaluate each one.
In Part 3, you will develop one high-fidelity prototype for part of your system and
prepare a formal evaluation plan for how you would assess its usability.
Homework 1 (8% total grade, group assignment): Simple evaluation of a software
application.
Students will carry out an informal evaluation of a software application, by asking
users outside your team to execute a set of tasks. You'll observe and document
problems you observe in terms of HCI principles.
Homework 2 (8% total grade, group assignment): Evaluation using Morae.
Students will carry out a more formal evaluation of a software application, by
observing and recording user-evaluation sessions using a new software tool called
Morae (http://www.techsmith.com/products/morae/)
Homework 3 (8% total grade, individual assignment): Developing a GUI.
Students use a language of their choice (C++ with Qt, Java, perhaps Visual Basic) and
create an application with a windowed user-interface that uses menus, buttons,
multiple windows, etc.

Topics Covered

What Is Interaction Design and HCI?
and Establishing Requirements, Understanding and Conceptualizing Interaction,
Understanding Users: Cognition, Psychology, Design and Prototyping, Physical UI

Prerequisites by Topic

- Software development maturity equivalent to two programming courses
- Introduction to software engineering lifecycles and process.
- Experience in a language to be able to learn to create a user-interface using a
  GUI library or toolkit in a language like C++ or Java.)
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Advanced Software Development Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CS 3240</strong></td>
<td>Course Title</td>
<td>Sherriff</td>
</tr>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td>Date of Preparation</td>
</tr>
</tbody>
</table>

Current Catalog Description

Analyzes modern software engineering practice for multi-person projects; methods for requirements specification, design, implementation, verification, and maintenance of large software systems; advanced software development techniques and large project management approaches; project planning, scheduling, resource management, accounting, configuration control, and documentation. Prerequisite: CS 216 with a grade of C- or higher.

Textbook


Course Outcomes

1. Develop an understanding of how to specify, design, and implement a complex software entity that involves many aspects of modern software systems.
2. Master a number of modern tools and a number of difficult technical fields including user interface design, distributed architecture development, and concurrent programming.
3. Develop the skills necessary to prepare professional quality technical documents as are required in the creation of complex software systems.
4. Develop the presentation skills necessary to convey elaborate and detailed technical concepts to colleagues.
5. Acquire experience of working on a large software system as a member of a group working on system development and as a member of a group that has to interact with other groups and customer representatives.
**Class/Laboratory schedule**

<table>
<thead>
<tr>
<th>Laboratory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Laboratory 1</td>
<td>Environment</td>
</tr>
<tr>
<td>1 Laboratory 2</td>
<td>Risk Reduction Prototype</td>
</tr>
<tr>
<td>1 Laboratory 3</td>
<td>Requirements Analysis</td>
</tr>
<tr>
<td>1 Laboratory 4</td>
<td>Requirements Specification</td>
</tr>
<tr>
<td>1 Laboratory 5</td>
<td>Implementation Planning</td>
</tr>
<tr>
<td>1 Laboratory 6</td>
<td>Development Tool and Protocol</td>
</tr>
<tr>
<td>1 Laboratory 7</td>
<td>System Design</td>
</tr>
<tr>
<td>1 Laboratory 8</td>
<td>End-to-End Prototype</td>
</tr>
<tr>
<td>1 Laboratory 9</td>
<td>System Performability</td>
</tr>
<tr>
<td>1 Laboratory 10</td>
<td>Preliminary Implementation and Redesign</td>
</tr>
<tr>
<td>1 Laboratory 11</td>
<td>Inspections</td>
</tr>
<tr>
<td>1 Laboratory 12</td>
<td>Delivery</td>
</tr>
</tbody>
</table>

**Relationship between Course Outcomes and Program Outcomes**

- project management, scheduling, planning
- requirements capture and analysis
- semi-formal specification
- object-oriented design
- programming practices
- software reuse--application generator and component-based
- inspections
- formal specification

**Prerequisites by Topic**
### COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Principles of Software Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 4240</td>
<td>Course</td>
<td>Coordinator: Bloomfield</td>
</tr>
<tr>
<td>Semester hours</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation: Fall 2009</td>
</tr>
</tbody>
</table>

### Current Catalog Description

Focuses on techniques for software design in the development of large and complex software systems. Topics will include software architecture, modeling (including UML), object-oriented design patterns, and processes for carrying out analysis and design. More advanced or recent developments may be included at the instructor's discretion. The course will balance an emphasis on design principles with an understanding of how to apply techniques and methods to create successful software systems. Prerequisite: CS 216 with grade of C- or higher.

### Textbook

Due to the high cost of textbooks, we will not have a required or recommended textbook for this course. Instead, there will be required readings that will be provided electronically. Your favorite programming language reference book(s), or online reference(s), are highly recommended. We will be studying from the book 'Design Patterns' (ISBN 0201633612) quite extensively, but content remarkably similar to that book is available online at http://www.oodesign.com/

### Course Outcomes

Students who complete the course will:
- Create a requirements model using UML class notations and use-cases based on statements of user requirements, and to analyze requirements models given to them for correctness and quality.
- Create the OO design of a system from the requirements model in terms of a high-level architecture description, and low-level models of structural organization and dynamic behavior using UML class, object, and sequence diagrams.
- Comprehend enough OOP in at least one OO language to see how to create software the implements the OO designs modeled using UML.
- Comprehend the nature of design patterns by understanding a small number of
examples from different pattern categories, and to be able to apply these patterns in creating an OO design.

- Given OO design heuristics, patterns or published guidance, evaluate a design for applicability, reasonableness, and relation to other design criteria.

Class/Laboratory schedule

| TuTh 3:30PM - 4:45PM |

Prerequisites by Topic

| Programming and Data Representation (CS 2150) |
COURSE DESCRIPTION

Dept., Number | Course Title | Advanced Computer Architecture
---|---|---
CS 4330 | | |

Semester hours | Course Coordinator | Skadron
---|---|---
3 | |

Required/Elective Course | Date of Preparation | Jan. 2009
---|---|---
Elective | |

Current Catalog Description

Provides an overview of modern microprocessor design. The topics covered in the course will include the design of super-scalar processors and their memory systems, and the fundamentals of multi-core processor design. Prerequisite: CS 216 and CS/ECE 333, both with grades of C- or higher.

Textbook


Course Outcomes

1. Ability to describe the operation, purpose, and implementation of the main microarchitectural features in cutting edge processor cores
2. Ability to describe the rationale for and implications of multicore organizations for hardware and software
3. Ability to describe the operation, purpose, and implementation of common architectural features supporting parallel execution, as well as their software implications
4. Ability to describe how the architectural concepts examined in this course are deployed in state-of-the-art products
5. Ability to write short parallel programs that take advantage of the architectural features examined in this course

Class/Laboratory schedule

Class: Tu-Th 3:30
Prerequisites by Topic

Programing and Data Representation (CS 2150) and Computer Architecture (CS/ECE 3330), both with grades of C- or higher.
COURSE DESCRIPTION

Dept., Number | CS 4414 | Course Title | Operating Systems
--- | --- | --- | ---
Semester hours | 3 | Course Coordinator | Grimshaw
Required/Elective Course | Required | Date of Preparation | Spring 2010

Current Catalog Description

Analyzes process communication and synchronization; resource management; virtual memory management algorithms; file systems; and networking and distributed systems. Prerequisite: CS 216 and CS 333 with grades of C- or higher.

Textbook

Silberschatz, Galvin, Gagne, "Operating System Concepts" 8th edition

Course Outcomes

* Operating system structures
* Processes
* Inter-process communication
* Thread
* Scheduling
* Synchronization and deadlock
* The storage hierarchy and memory management
* Input/output
* File systems
* Security and protection
* Distributed systems basics

Class/Laboratory schedule

Tues/Thurs 12:30-13:45

This course has no labs, but requires four programming projects:
1. Build a simple shell (illustrates process creation and management) – 1 week
2. Build a simple kernel from initial skeleton code, using the NACHOS instructional operating system package (illustrates kernel organization, system calls, and interrupt handling) – 2 weeks
3. Reconstruct files from a corrupted disk (illustrates file-system organization) – 1 week
4. Break a supposedly secure program (illustrates security issues) – 1 week

Prerequisites by Topic

| Data Structures, Computer Architecture |
Introduces the student to the basics of high-performance parallel computing and the national cyber-infrastructure. The course is targeted for both computer science students and students from other disciplines who want to learn how to significantly increase the performance of applications.

Michael Quinn, “Parallel Programming in C with MPI and OpenMP”

Topics to be covered include: Performance measurement; single CPU performance optimization; Parallel computer architectures and Flynn’s taxonomy; Interconnection networks; Vector & SIMD processors; Multiprocessors; Multicomputers and clusters; Tightly coupled MPP’s; GPUs; Limits to parallelization, performance prediction; Embarrassingly (pleasingly parallel) applications; Data parallel applications, MPI, and OpenMP; Simple stencil problems; More complex irregular structures; Load balancing data parallel problems; Compiler techniques for automatic parallelization.

It is expected that the student know one or more of C, C++, or Fortran. Further, the student is expected to know rudimentary computer architecture, the role of the CPU, the memory system, and the caches.
COURSE DESCRIPTION

Dept., Number | CS 4620 | Course Title | Compilers
--- | --- | --- | ---
Semester hours | 3 | Course Coordinator | Hazelwood
Required/Elective Course | Elective | Date of Preparation | Spring 2010

Current Catalog Description

Provides an introduction to the field of compilers, which translate programs written in high-level languages to a form that can be executed. The course covers the theories and mechanisms of compilation tools. Students will learn the core ideas behind compilation and how to use software tools such as lex/flex, yacc/bison to build a compiler for a non-trivial programming language.

Textbook


Course Outcomes

- Structure of Compilers
- Lexical Analysis
- Parsers
- Attribute Grammars, Actions, and Parse Trees
- Scoping in Language Definition
- Symbol Tables
- Type Checking
- Run-time Data Structures
- Intermediate Code
- Code Generation
- Dataflow Analysis
- Liveness Analysis and Register Allocation

Class/Laboratory schedule

Tues/Thurs 2:00-3:15

Prerequisites by Topic

Advanced Software Development Techniques and Digital Logic Design. A strong programming background will be important to your success in this course
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 4710</th>
<th>Course Title</th>
<th>Artificial Intelligence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Worthy Martin</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>Spring 2009</td>
</tr>
</tbody>
</table>

Current Catalog Description

Introduces artificial intelligence. Covers fundamental concepts and techniques and surveys selected application areas. Core material includes state space search, logic, and resolution theorem proving. Application areas may include expert systems, natural language understanding, planning, machine learning, or machine perception. Provides exposure to AI implementation methods, emphasizing programming in Common LISP.

Textbook


References

http://www.cs.virginia.edu/~cs416/

Course Outcomes

To acquire a broad appreciation for search and optimization:

1. Identification of state space
2. Identification of action space
3. Logic and inference
4. Identification of search/optimization techniques (informed, local, global, constrained, adversarial)
5. Learning from observation (stochastic, model building, neural networks, reinforcement learning)
Class/Laboratory schedule

1. Informed search in discrete spaces – solve the 15-puzzle using six different search techniques and heuristics (3 weeks).
3. Learning from observation – Gambler’s Ruin problem from Reinforcement Learning text by Sutton and Barto (2 weeks).

Relationship between Course Outcomes and Program Outcomes

Agents, Uninformed Searches, Informed Searches, Adversarial Search, Logical Agents, Propositional Logic, First-Order Logic, Neural Networks, Biologically Inspired Neural Networks, Making Complex Decisions, Statistical Learning, Hidden Markov Models

Prerequisites by Topic

Calculus required. Basic linear algebra and statistics is recommended, but will be reviewed in class. As this course is intended for upper-class computer science majors, the CS 216 prerequisite represents a minimal amount of programming skills. The programming will require significant programming efforts.
Current Catalog Description

Introduces the fundamental concepts for design and development of database systems. Emphasizes relational data model and conceptual schema design using ER model, practical issues in commercial database systems, database design using functional dependencies, and other data models. Develops a working relational database for a realistic application.

Textbook


References

*Database systems (4th edition)*, Elmasri and Navathe, Addison Wesley

Course Outcomes

The students will learn design concepts for database using ER modeling, and relational database schema construction. Some of the fundamental concepts of relational database systems will be covered, including functional dependencies and normal forms. Database languages, including SQL will be discussed to certain level of detail so that students can compose queries with desired functionalities using SQL. System issues of database management systems including storage management, concurrency control and recovery, indexing, and data and query distribution are also introduced.
Class/Laboratory schedule

Design and implementation of relational database (term project)

Relationship between Course Outcomes and Program Outcomes

1. ER model
2. SQL
3. Database integrity
4. relational database design
5. storage and file structures
6. indexing
7. query processing
8. transactions
9. concurrency control
10. database recovery

Prerequisites by Topic

Programming proficiency in Java/C++
Introduction to software development methodologies
Data structures
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Electronic Commerce Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 4753</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Semester hours 3

Course Coordinator Alfred Weaver

Required/Elective Course Elective

Date of Preparation June 23, 2010

Current Catalog Description

History of Internet and electronic commerce on the web; case studies of success and failure; cryptographic techniques for privacy, security, and authentication; digital money; transaction processing; wired and wireless access technologies; JavaScript; streaming multimedia; XML; Bluetooth. Defining, protecting, growing, and raising capital for an e-business.

Textbook

none

References

real-time exploration of the Internet and World Wide Web

Course Outcomes

A. Understand the technical and business requirements of a successful e-commerce venture, including the fundamental roles of privacy, security, authorization, and trust.
B. Comprehend the fundamental mathematics of symmetric key and public key cryptography. Review DES, RSA, SSL, PGP, hash functions, digital signatures, and certification authorities.
C. Utilize HTML, CSS, JavaScript, CGI, SQL, PHP, mySQL and web services to develop significant operational e-commerce modules (e.g., establishing identity via a biometric device).
D. Understand and appreciate how intellectual property is developed, evaluated and evaluated; how it contributes to corporate valuation; how it is protected and how property rights are enforced via copyrights, trademarks, trade secrets, and patents.
E. Develop a substantial e-commerce enterprise: create a social networking website called Artist eXchange that supports client registration, mySQL database management, uploading files (images, movies, music), voting and commenting by
viewers, and website security that will defeat common attack vectors (e.g., buffer overflow, SQL injection).

Class/Laboratory schedule

Lecture: 3:30-4:45 Monday/Wednesday

Prerequisites by Topic

Two semesters of programming, and one semester of data structures.
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 1110 &amp; 1111</th>
<th>Course Title</th>
<th>Introduction to Computing</th>
</tr>
</thead>
</table>

Semester hours 3 Course Coordinator Horton, Cohoon

Required/Elective Course Required Date of Preparation 6/1/2009

Note on CS 1110/1111:
CS1111 (formerly CS101E) are normally offered each term (but were not in Fall 2009). CS1110 and CS1111 are in effect two sections of the same course. Students with at least a semester worth of prior formal class-work in programming in any language are strongly encouraged to take CS1111. Students without such prior experience are required to take CS1110. CS1110 and CS1111 will cover the same lecture material. Exams, assignments, and grading are the same for both sections. CS1111 students are not required to attend a closed lab section.

Current Catalog Description

Introduces the basic principles and concepts of object-oriented programming through a study of algorithms, data structures and software development methods in Java. Emphasizes both synthesis and analysis of computer programs.

Textbook


References

On-line information found on the textbook authors' “booksite” at http://www.cs.princeton.edu/introcs

Course Outcomes

Upon completion of CS1110 or CS1111, a student will:

- Understand common fundamentals of programming such as variables, conditional and iterative execution, functions, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, calling member functions, using class libraries, etc.
- Have an appreciation of important topics and principles of software development and computer science.
- Be able to write a computer program to solve a specified problem.
- Have strong practical experience using the Java programming environment.
to create, debug and run simple Java programs.

Class/Laboratory schedule

The lecture schedule follows the list of high-level topics described below. Students in CS1110 must attend a weekly lab for 75 minutes, where they work alone or with partners on exercises. The Fall 2009 lab schedule follows. (Three labs sessions were devoted to programming quizzes.)
Lab 1: Introduction to programming environment
Lab 2: Exercises using simple programs
Lab 3: Loops and conditional statements
Lab 4: Arrays. Also, practice for Lab Quiz
Lab 5: Using the textbook's graphics drawing library
Lab 6: Functions
Lab 7: Strings objects, DNA example
Lab 8: Color and Picture objects
Lab 9: Creating new datatypes with classes

Topics Covered

- Course Introduction
- First Program
- Loops
- Arrays
- Input and Output
- Functions and Libraries
- Recursion
- Algorithms
- Using Data Types
- Creating Data Types
- ADTs
- Stacks and Queues
- Symbol Tables

Prerequisites by Topic

None for CS1110.

For CS1111, at least one semester’s worth of programming in any language.
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>CS 2110</th>
<th>Course Title</th>
<th>Software Development Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Horton</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td>Date of Preparation</td>
<td>6/15/10</td>
</tr>
</tbody>
</table>

Current Catalog Description

A continuation of CS 101, emphasizing modern software development methods. An introduction to the software development life cycle and processes. Topics include requirements analysis, specification, design, implementation, and verification. Emphasizes the role of the individual programmer in large software development projects. Prerequisite: CS 101 with a grade of C- or higher. Note: Students may not receive credit for both CS 201 and CS 205.

Textbook


Course Outcomes

1. Comprehend more advanced principles of object-oriented programming and how a programming language supports these, and apply these by developing larger and more complex programs than in their first programming course. (Topics include polymorphism, inheritance, C++ templates, etc.)
2. Comprehend and apply principles of design at the class and object level. These principles include abstraction, encapsulation, and information hiding. This also includes the ability to define and evaluate class interfaces to solve specified design problems, as well as the ability to understand, apply, and evaluate the use of reusable components to solve such problems.
3. Comprehend and analyze problems and programming issues such as dynamic memory management, pointers and other indirect object references, and recursion. Also, be able to apply this knowledge by implementing software that includes these features.
4. Apply knowledge of software development practice to effectively use tools environments such as interactive development environments, debuggers, etc.
5. Comprehend important basic concepts of software engineering and the
development of large software systems, including the software lifecycle, requirements, design, and software quality. In their development activities, students will be able to apply basic unit testing and carry out a software inspection.

6. Comprehend the basic principles of the architecture of larger software systems, in particular object-oriented frameworks. Apply this knowledge by developing a small program using a framework.

Class/Laboratory schedule

**Lab1: Warm-Up Activities**
Your first, official lab. Work through some I/O activities using the Scanner, and then work on HW0.

**Lab2: You are your own worst enemy...**
In this lab, you and your partner will learn to work with the debugger built into Eclipse.

**Lab3: Second Grade Grammar**
In this lab, you and your partner will do some basic class design and diagramming.

**Lab4: Complaining about testing doesn’t make it go away...**
In this lab, you and your partner will learn the intricacies of JUnit.

**Lab5: A Prelude to Bacon**
In this lab, you and your partner will get to know the JCF a little better.

**Lab6: Real Estate**
In this lab, you and your partner will perform maintenance on a GUI driven program, in anticipation of the project.

**Lab7: Trees!**
In this lab, you and your partner will begin working with trees.

**Lab8: Project Week 1**
In this lab, you will meet with your team to determine team roles and begin working on your first RAF.

**Lab9: Project Week 2**
In this lab, teams will present their first RAF to the TAs and then continue work on the project.

**Lab10: Project Week 3**
In this lab, teams will present their second RAF to the TAs and then continue work on the project.

**Lab11: Project Week 4 - The final week!**
In this lab, you will work on finishing your project for demoing next week!

**Lab12: Project Demos**
In this lab, you will show off your hard work!

**Lab13: Threading War**
In this lab, you will begin to build the card game War with threads.

**Lab14: Course Review**
In this lab, you will review the past few months.

Prerequisites by Topic

Basic working knowledge of Java (up through creating classes)
### COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Dependent Computer Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 4340</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester hours</th>
<th>Course Coordinator</th>
<th>Joanne Bechta Dugan</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required/Elective Course</th>
<th>Date of Preparation</th>
<th>Joanne Bechta Dugan, May 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elective</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Current Catalog Description**

Focuses on the techniques for designing and analyzing dependable computer-based systems. Topics include fault models and effects, fault avoidance techniques, hardware redundancy, error detecting and correcting codes, time redundancy, software redundancy, combinatorial reliability modeling, Markov reliability modeling, availability modeling, maintainability, safety modeling, trade-off analysis, design for testability, and the testing of redundant digital systems. Cross-listed as ECE 434. Prerequisite: CS 333, APMA 213, APMA 310, with grades of C- or higher, or instructor permission.

**Textbook**


**References**

Many readings from current literature will be posted on the collab site and discussed in class.

**Course Outcomes**

Upon successful completion of this course, students will be able to:

- **Objective 1.** analyze a computer-based system with respect to non-functional attributes, such as reliability, availability, yield, error detection, etc.
- **Objective 2.** utilize redundancy in time, space, information and hardware to prevent, detect and recover from errors and the faults that produce them.
Objective 3. analyze the dependability of a computer-based system using probability models
Objective 4. reason about tradeoffs between dependability, cost, performance, complexity, etc.

The specific topics to be covered may be influenced by the interests of the class attendees and may include:
- Hardware Fault Tolerance
- Information Redundancy
- Software Fault Tolerance
- Fault Tolerant Networks
- Software Fault Tolerance
- Checkpointing
- Fault Detection in Cryptographic Systems
- Defect Tolerance and Soft errors

Class/Laboratory schedule

3 50-minute class meetings per week

Relationship between Course Outcomes and Program Outcomes

Contributes to outcome (e) an ability to identify, formulate and solve engineering problems

Prerequisites by Topic

ECE/CS 3330 Computer Architecture, APMA 3100 Probability and APMA 2130 Differential Equations
Current Catalog Description

A first course in communication networks for upper-level undergraduate students. Topics include the design of modern communication networks; point-to-point and broadcast network solutions; advanced issues such as Gigabit networks; ATM networks; and real-time communications.

References


Course Outcomes

• Principles:
  – How do switches work? How do networks work?
  – Enable you to design the next-generation of switches and networks
• Practice:
  – How does the Internet work?
  – How does the telephone network work?

Class/Laboratory schedule

Meet twice weekly for class lectures (T/R 12:30 to 1:45)
Relationship between Course Outcomes and Program Outcomes

| Principles: The students will learn different general techniques for networking tasks, such as error control, flow control, switching and routing. |
| 2. Practice: The students will gain an understanding of how existing and next-generation communication networks work. |

Prerequisites by Topic

| Computer architecture |
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE/CS 2330</td>
<td>Digital Logic Design</td>
<td>Joanne Bechta Dugan</td>
</tr>
</tbody>
</table>

Semester hours 3  

Required/Elective Course Required

Date of Preparation May 2010

Current Catalog Description

Includes number systems and conversion; Boolean algebra and logic gates; minimization of switching functions; combinational network design; flip-flops; sequential network design; arithmetic networks. Five laboratory assignments.

Textbook


Course Outcomes
**Course Learning Outcomes**

Upon successful completion of this course, students will be able to:

- **Objective 1.** analyze and design combinational digital circuits.
- **Objective 2.** analyze and design sequential digital circuits.
- **Objective 3.** implement combinational and clocked sequential digital circuits.
- **Objective 4.** design simple systems at the Register Transfer Level (RTL Design)

**Intangible Goals**

After taking this course, a student will:

- **Goal 1:** better understand how digital systems dramatically impact people's lives
- **Goal 2:** better understand how electrical and computer scientists and engineers build digital systems
- **Goal 3:** have improved problem-solving skills, including laboratory and debugging skills
- **Goal 4:** better understand multiple levels of abstraction, that is, the ability to move from detailed component-level design to system design (where components are treated as building blocks) and vice-versa
- **Goal 5:** better understand the need for precision in technical communications, particularly with respect to interface design

**Class/Laboratory schedule**

3 50-minute classes per week; 2-hour lab alternate weeks; optional recitation 1.5 hours

**Relationship between Course Outcomes and Program Outcomes**

The *ability to apply knowledge of math and science to engineering problems* (outcome (a)) is assessed in this course.

**Prerequisites by Topic**

None

**Criterion 5 Contribution:** Engineering science; Engineering Design
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE/CS 3330</th>
<th>Course Title</th>
<th>Computer Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester Hours</td>
<td>3 Required for EE CPE and CS</td>
<td>Course Coordinator</td>
<td>John Lach</td>
</tr>
</tbody>
</table>

Current Catalog Description

Includes the organization and architecture of computer systems hardware; instruction set architectures; addressing modes; register transfer notation; processor design and computer arithmetic; memory systems; hardware implementations of virtual memory, and input/output control and devices. Cross-listed as ECE 3330 and CS 3330.

Textbook


Course Outcomes

**Objective 1.** To provide students a strong background in the organization and architecture of computer systems hardware.

**Objective 2.** To introduce the primary components of a computer, including processor, memory, input, and output.

**Objective 3.** To expose students to commercial processors as examples of the technology.

**Objective 4.** To gain practical experience with computers through assembly language programming on a representative processor and through the design of a simple processor architecture.

Prerequisites by Topic

CS 2110 (Software Development Methods) or 2220 with a grade of C- or higher, and ECE/CS 2330 (Digital Logic Design) with a grade of C- or higher.

Major Topics Covered in the Course
* History of computers
* Review of digital logic design
* Basic structure of computer hardware and software
* Computer instruction sets
* Assembly language programming
* Addressing modes
* Example processors
* Processor data path
* Hardwired control units
* Microprogrammed control units
* Pipelining
* Computer arithmetic
* Memory components
* Memory systems
* Input and output units
* Interrupts
* Direct memory access (DMA)
* Peripheral devices
* Computer communications
* Computer networks
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 2630</th>
<th>Course Title</th>
<th>Introductory Circuit Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Holmes</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td>Date of Preparation</td>
<td>June 12, 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Elementary electrical circuit concepts and their application to linear circuits with passive elements; use of Kirchhoff’s voltage and current laws to derive circuit equations; solution methods for first- and second-order transient and DC steady-state responses; AC steady-state analysis; frequency domain representation of signals; trigonometric and complex Fourier series; phasor methods; complex impedance; transfer functions and resonance; Thevenin/Norton equivalent models; controlled sources.

Textbook


Course Outcomes

1. Define current, voltage, energy, and power and their relationships with each other.
2. Be able to solve for DC and AC voltages, currents, energies, and powers in linear circuits using the following techniques: Kirchhoff’s Laws; voltage and current division; node voltage analysis; mesh current analysis; and superposition.
3. Be able to analyze and design circuits which use op-amps, capacitors, inductors, and other devices.
4. Be able to build a circuit on a breadboard based on a circuit schematic and
accurately measure currents and voltages from this circuit

Class/Laboratory schedule

5 Laboratory assignments

Relationship between Course Outcomes and Program Outcomes

<table>
<thead>
<tr>
<th>Objective</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply mathematics science and engineering principles</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to design and conduct experiments and interpret data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to design a system, component, or process to meet desired needs</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ability function on multidisciplinary teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to identify, formulate, and solve engineering problems</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to understand professional and ethical responsibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to communicate effectively</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ability to understand the impact of engineering solutions in a global context</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to recognize the need for and to engage in life-long learning</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ability to Know of contemporary issues</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Prerequisites by Topic

Single Variable Calculus
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 2660</th>
<th><strong>Course Title</strong></th>
<th>Electronics I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>4</td>
<td><strong>Course Coordinator</strong></td>
<td>Blalock</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td><strong>Date of Preparation</strong></td>
<td>6/11/10</td>
</tr>
</tbody>
</table>

Current Catalog Description

Studies the modeling, analysis, design, computer simulation, and measurement of electrical circuits which contain non-linear devices such as junction diodes, bipolar junction transistors, and field effect transistors. Includes the gain and frequency response of linear amplifiers, power supplies, and other practical electronic circuits.

Textbook


Course Outcomes

1. Students will be familiar with the basic semiconductor principles of active device operation
2. Students will be able to analyze and predict the behavior of basic electronic circuits.
3. Students will be able to design, simulate, and test practical electronic circuits.

Class/Laboratory schedule

Three lecture and three laboratory hours

Prerequisites by Topic

*Introductory Circuit Analysis*
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 3103</th>
<th>Course Title</th>
<th>Physics of Semiconductor Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester Hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Lloyd Harriott</td>
</tr>
</tbody>
</table>

Current Catalog Description

Today, there are more transistors than ants on Earth! A single semiconductor factory can produce about a million transistors for every person on the face of the earth every week! This course will enable you to understand what transistors are and how they work along with other related semiconductor devices. We start by introducing semiconductor materials and describe how they can be used to make useful electronic devices. We will necessarily introduce some basic concepts in Quantum Mechanics to explain the operation of semiconductors (although no prior knowledge of QM is assumed). Once we have understood semiconductor materials, we will then describe the operation of the basic semiconductor building blocks and devices including diodes and transistors. This course is intended for undergraduate students who have an interest in microelectronics. It is expected that students who successfully complete the course will have an understanding of basic semiconductor devices sufficient to design transistors and diodes to particular specifications.

Textbook


Course Outcomes

Objective 1. Analyze and design semiconductor devices including diodes, MOSFET and BJT.
Objective 2. Draw energy band diagrams for any semiconductor device

Prerequisites by Topic

College level calculus including ordinary differential equations is required. A basic course in circuits is helpful but not required
Major Topics Covered in the Course

**Topics Covered**

- Crystals and Semiconductor Materials
- Introduction to Quantum Mechanics (QM101)
- Application to Semiconductor Crystals – Energy Bands
- Carriers and Statistics
- Recombination-Generation Processes
- Carrier Transport Mechanisms
- P-N Junctions
- Metal-Semiconductor Contacts – Schottky Diodes
- Metal-Oxide-Semiconductor Transistor (MOSFET)
- MOSFET Operation and Scaling
- Bipolar Junction Transistors (BJT)
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 3209</th>
<th>Course Title</th>
<th>Electromagnetic Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Weikle</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective for CpE Required for EE</td>
<td>Date of Preparation</td>
<td>June 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Analyzes the basic laws of electromagnetic theory, beginning with static electric and magnetic fields, and concluding with dynamic E&M fields; plane wave propagation in various media; Maxwell’s Laws in differential and integral form; electrical properties of matter; transmission lines, waveguides, and elementary antennas.

Textbook

Field and Wave Electromagnetics, 2nd ed. By David K. Cheng

References

Schaum’s Outline of Electromagnetics, 2nd ed. By Joseph Edminister

Course Outcomes

To explain, through lectures and assignments, the fundamental principles of electromagnetic field theory.
To deepen students’ understanding of vector analysis and multivariable calculus and their application to electromagnetic
To introduce students to the basic techniques for applying electromagnetic field theory to solving practical problems of interest to engineering.

Class/Laboratory schedule
Lectures on Mondays and Wednesdays and problem recitation/discussion on Friday.

Prerequisites by Topic

Physics II, Ordinary Diff. Equations, Introductory Circuit Analysis
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 3250</th>
<th>Course Title</th>
<th>Electromechanical/Electromagnetic Energy Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Harry Powell</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description


Textbook

Electric Machinery and Power Systems Fundamentals by Chapman

Course Outcomes

At the end of this course students should be able to:
1. Analyze single-phase and three-phase sinusoidal steady state circuits containing transformers, generators, and motors.
2. Analyze flux and flux generation in magnetic structures
3. Analyze performance of ac induction motors.
4. Experimentally determine the equivalent circuit model of a transformer and induction motor.
5. Understand the basics of switching power supplies.
6. Understand the use of electromagnetic energy conversion in sensors and other specialized applications

Prerequisites by Topic

ECE 203, PHYS 241E, (CS 101, ECE 309, ECE 204 are recommended)
1. AC phasor transform circuit analysis 2. Electromagnetic Fields 3. Computer programming
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 3632</th>
<th>Course Title</th>
<th>Electronics II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Blalock</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>6/11/10</td>
</tr>
</tbody>
</table>

Current Catalog Description

Construction of electronic circuit design to specifications. Focuses on computer simulation, construction, and testing of designed circuits in the laboratory to verify predicted performance. Includes differential amplifiers, feedback amplifiers, multivibrators, and digital circuits.

Textbook


Course Outcomes

1. Students will be able to analyze DC bias and small-signal characteristics of multi-stage transistor amplifiers.
2. Students will be able to analyze frequency response of open and closed loop amplifier circuits.
3. Students will be able to design, simulate, and experimentally evaluate practical feedback amplifiers.

Class/Laboratory schedule

Three Lecture Hours and Three Lab Hours

Prerequisites by Topic

Electronics I
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 3750</th>
<th>Course Title</th>
<th>Signals and Systems I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Brandt-Pearce</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Required</td>
<td>Date of Preparation</td>
<td>6/18/2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Develops tools for analyzing signals and systems operating in continuous-time, with applications to control, communications, and signal processing. Primary concepts are representation of signals, linear time-invariant systems, Fourier analysis of signals, frequency response, and frequency-domain input/output analysis, the Laplace transform, and linear feedback principles. Practical examples are employed throughout, and regular usage of computer tools (Matlab, CC) is incorporated.

Textbook


Course Outcomes

1. understand concepts of signals and systems
2. transform between time domain and frequency domain representations of signals
3. model and analyze LTI systems in both time and frequency domains
4. use MATLAB to visualize signals, and to perform simple system simulations

Class/Laboratory schedule

Three 50 minute lectures plus two one-hour recitation sessions.

Prerequisites by Topic

Introductory Circuit Analysis and Ordinary Diff. Equations
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 3760</th>
<th>Course Title</th>
<th>Signals and Systems II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Brandt-Pearce</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>6/18/2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Sequel to ECE 323; provides analogous tools for analyzing discrete-time signals and systems, with applications to discrete-time signal processing and control. Sampling and reconstruction of continuous-time signals provides the transition between CT and DT settings. State space methods are also introduced.

Textbook


Course Outcomes

1. transform between time domain and frequency domain representations of signals
2. model and analyze LTI systems in both time and frequency domains
3. use MATLAB to visualize signals, and to perform simple system simulations
4. apply transform methods to circuits and data analysis problems

Class/Laboratory schedule

Three 50 minute lectures plus a one-hour recitation session.

Prerequisites by Topic

Signals and Systems I
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Introduction to VLSI Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE 4332</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester hours</th>
<th>Course Coordinator</th>
<th>Date of Preparation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Calhoun</td>
<td>June 10, 2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required/Elective Course</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Elective (counts as MDE for EE)</td>
<td></td>
</tr>
</tbody>
</table>

Current Catalog Description

In this course, you will learn how to analyze, simulate, and design ready-to-fabricate digital integrated circuits (ICs). Topics include combinational circuits, sequential circuits, arithmetic blocks, and memories. We will discuss modern VLSI design issues including variation, power, leakage, and optimization. A semester-long design project (small groups) will allow you to apply your new knowledge to your own IC design. You will gain valuable experience by using industry-standard CAD tools for VLSI circuit design, simulation, and layout during the project.

Textbook


Course Outcomes

Upon successful completion of this course, students will be able to:

**Objective 1.** Understand static, dynamic, sequential, memory, variation and power

**Objective 2.** Know how to approach and analyze an unfamiliar circuit

**Objective 3.** Know how to implement a chip from end to end

a. know how to identify key tradeoffs for a spec
b. reason about what circuits to use
c. schematic, simulation, layout, verification
d. know how to partition a design and work with a team (hierarchical circuit design)
**Objective 4.** Know how to use Cadence and how to learn about new CAD tools  
**Objective 5.** Learn how to assess your own circuit analysis and design  
**Objective 6.** Learn how to run down and solve a problem

Class/Laboratory schedule

TR 2:00pm – 3:15pm

Relationship between Course Outcomes and Program Outcomes

Assessment of MDE (major design experience) is performed in this course

Prerequisites by Topic

COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Semester hours</th>
<th>Course Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE 4435</td>
<td>Computer Organization and Design</td>
<td>4.5</td>
<td>Williams</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required/Elective Course</th>
<th>Date of Preparation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required for CpE</td>
<td></td>
</tr>
<tr>
<td>Elective for EE</td>
<td></td>
</tr>
</tbody>
</table>

Current Catalog Description

Integration of computer organization concepts, such as data flow, instruction interpretation, memory systems, interfacing, and microprogramming with practical and systematic digital design methods such as behavioral versus structural descriptions, divide-and-conquer, hierarchical conceptual levels, trade-offs, iteration, and postponement of detail. Design exercises are accomplished using a hardware description language and simulation.

Course Outcomes

1. Students will be familiar with VHDL and Computer-Aided Digital Design
2. Students will be able to undertake design of complex synchronous digital systems
3. Students will be familiar with advanced computer architecture techniques used for hardware acceleration.

Class/Laboratory schedule

Lectures meet twice weekly for 1 hour and 15 minutes per meeting.
Laboratories meet 11 times during the semester for 3 hours per meeting.

Prerequisites by Topic

Basic Digital Logic Design
Basic Computer Architecture
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 4440</th>
<th>Course Title</th>
<th>Advanced Digital Design</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Semester hours</th>
<th>4.5 (counts as MDE)</th>
<th>Course Coordinator</th>
<th>Stan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required/Elective Course</td>
<td>Required for CpE/Elective for ECE/CS</td>
<td>Date of Preparation</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Analyzes digital hardware and design; digital system organization; digital technologies; and testing. A semester-long hardware design project is conducted.

Textbook

ASICs - The Book (hardcopy or on-line)
http://www.edacafe.com/books/ASIC/ASICs.php
Michael John Sebastian Smith, Addison-Wesley, ISBN: 0-201-50022-1

References

OpenSPARC Internals companion book is available here:

Course Outcomes

This course extends the theory and practice of digital system design that was introduced in earlier courses. The digital design theory introduced in ECE/CS 230: Digital Logic Design, ECE/CS 333: Computer Architecture, and ECE 435: Computer Organization and Design will be complemented with practical aspects that will be immediately useful in implementing a simple computer system based on an FPGA implementation of a custom processor. It is imperative that you have taken and passed 230, 333, and 435 in order to be able to take ECE436. Since "doing" is a major aspect of engineering education, the laboratory is a significant portion of this course.

Class/Laboratory schedule
While there are no formal lab periods for this class (as the CAD software is available on all ITC Stacks machines), each student will have a priority lab day. That is, a student will have priority access to TAs, machines, and equipment during TA office hours on his/her priority lab day. While students are not required to attend their laboratory session (except for laboratory assignments that require check-offs – see below), it will be the best opportunity to get in-depth TA assistance.

Prerequisites by Topic

Computer Organization and Design
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Number</th>
<th>Course Title</th>
<th>Required/Elective</th>
<th>Semester hours</th>
<th>Course Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE 4710</td>
<td></td>
<td>Communications</td>
<td>Elective</td>
<td>3</td>
<td>Wilson</td>
</tr>
</tbody>
</table>

Current Catalog Description

Explores the statistical methods of analyzing communications systems: random signals and noise, statistical communication theory, and digital communications. Analysis of baseband and carrier transmission techniques; and design examples in satellite communications.

Textbook

Digital and Analog Communication Systems, L. Couch, 7th Ed.

Course Outcomes

To provide familiarity with the probabilistic concepts of communications theory: to acquaint students with basics of communication link calculations; to provide an understanding of the fundamentals of analog and digital communications; and to bring student to current state of technology in aspects of communication engineering via class presentations by practitioners.

Class/Laboratory schedule

Mon, Wed. Fri 12:00PM - 12:50PM

Prerequisites by Topic

APMA 310 and Signals and Systems II
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE 4784</td>
<td></td>
<td>Wireless Communications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semester hours</th>
<th>Course Coordinator</th>
<th>Date of Preparation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Wilson</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

This is a survey course in the theory and technology of modern wireless communication systems, exemplified in cellular telephony, paging, microwave distribution systems, wireless networks, and even garage door openers. Wireless technology is inherently interdisciplinary, and the course seeks to serve the interests of a variety of students.

Textbook

Modern Wireless Communications, S. Haykin and M. Moher, Person/Prentice-Hall

References

Principles of Mobile Communications, G. Stuber, Kluwer
Wireless Communications, T. Rappaport, Prentice-Hall, 2nd Ed.
Wireless Communications, A. Goldsmith, Cambridge (on reserve in Sci-Engr Library)

Course Outcomes

Primary emphasis will be upon system capacity calculations for cellular systems; propagation calculations; digital system performance; and important standards, including cellular, wireless LAN and so-called personal area networks.

Prerequisites by Topic

Understanding of signals and systems calculus in continuous and discrete-time; basic notions of probability and random processes. This material is typical of that found in APMA 310 (old), ECE 3752/3754, ECE 4710, and ECE 576 (old). Skill with math packages such as Matlab or MathCad will be eventually needed. If you are unsure of your background, please contact me.
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ECE 4850</th>
<th>Course Title</th>
<th>Linear Control Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Gang Tao</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

Explores the modeling of linear dynamic systems via differential equations and transfer functions utilizing state space representations and classical input-output representations; the analysis of systems in the time and frequency domains; study of closed-loop systems; state-space methods and the classical stability tests, such as the Routh-Hurwitz criterion, Nyquist criterion, root-locus plots and Bode plots. Studies compensation design through lead and lag networks, rate feedback, and linear state-variable feedback.

Textbook


Course Outcomes

**Topics:**
1. Mathematical models of control systems (2 lectures)
2. Transfer functions and block diagrams (2 lectures)
3. Time-domain responses (2 lectures)
4. The Routh-Hurwitz stability criterion (2 lectures)
5. Feedback control systems (2 lectures)
6. Root-locus techniques (4 lectures)
7. Bode plot techniques (2 lectures)
8. The Nyquist stability criterion (2 lectures)
9. Dynamic compensation in frequency-domain (3 lectures)
10. State space analysis and design (4 lectures).

Class/Laboratory schedule

11:00 - 12:15, Tuesdays and Thursdays

Prerequisites by Topic

Signals and Systems
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Digital Control Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE 4860</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Semester | Hours | Course Coordinator |
----------|-------|--------------------|
3         |       | Zongli Lin         |

Current Catalog Description

Analyzes the design of dynamic systems that contain digital computers; the Z transform; block diagrams and transfer functions in the z-domain; block diagrams, frequency response and stability in the z-domain; state space methods; and design using the z-transform and state methods. Prerequisite: ECE 4850 or instructor permission.

Textbook


Course Outcomes

1. to carry out transform between time domain and frequency domain signals (z-transform, Laplace transform and starred transform) and to analyze signals;
2. to analyze a closed-loop system (stability, rise time, overshoot, settling time, steady state error, etc.);
3. to design controllers to the specifications.

Prerequisites by Topic

Linear Control Systems

Major Topics Covered in the Course

Introduction to discrete-time control systems and difference equations, z-transform, state space representation, data sampling and reconstruction, transform function in discrete-time, system time response characteristics, stability analysis techniques, digital controller design techniques.
COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>ENGR 2595</th>
<th>Course Title</th>
<th>Engineering in Community Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Paxton Marshall</td>
</tr>
<tr>
<td>Required/Elective Course</td>
<td>Elective</td>
<td>Date of Preparation</td>
<td>Spring 2010</td>
</tr>
</tbody>
</table>

Current Catalog Description

This course will provide students with an orientation to engineering within community settings, with particular emphasis on communities in developing areas outside of the United States, although underserved domestic communities will also be studied. The course will focus on equipping students with a broad range of technical and contextual knowledge through a combination of readings, case studies, discussions, guest presentations, group projects, and laboratory work. Students will study technologies aimed at meeting the basic human needs of water, air, food, shelter, household energy, and security. Topics will also include an overview of global health, research ethics and protocols, approaches to responsible and sustainable community engagement and instruction on grant writing.

The primary student activity of the course will be to design and begin implementation of a community project. Students will follow a structured design approach, consisting of Topic Selection, Problem Definition, Identification of Stakeholders and their needs, Project Specifications, Generation of Alternatives, Background Research on alternatives, Evaluation of Alternatives, Preliminary Design, Final Design. Students will be expected to produce a final project report, a final presentation to the class, and proposal for funding for the project. In addition there will be interim reports and reviews.

In previous offerings of this course, students have developed and implemented projects in South Africa, Belize, Nicaragua, Charlottesville, and elsewhere. Projects have been awarded funding from the Jefferson Public Citizens program, the Center for Global Heath, Davis Projects for Peace, and the Center for Undergraduate Excellence, among other funding sources.

This course is a component of the Jefferson Public Citizens program (JPC), which seeks to inspire students to act as engaged citizens through active community partnerships, research service projects, and scholarly reflection. It encourages and prepares students to work with local, national and international communities to effect positive change in the world.

Academic Community Engagement courses serve as cornerstones of the JPC program by preparing students to connect their academic experiences with public life and real world issues.
Course Outcomes

The course will focus on equipping students with a broad range of technical and contextual knowledge through a combination of readings, case studies, discussions, guest presentations, group projects, and laboratory work. Students will study technologies aimed at meeting the basic human needs of water, air, food, shelter, household energy, and security. Topics will also include an overview of global health, research ethics and protocols, approaches to responsible and sustainable community engagement and instruction on grant writing.

This course is a component of the Jefferson Public Citizens program (JPC), which seeks to inspire students to act as engaged citizens through active community partnerships, research service projects, and scholarly reflection. It encourages and prepares students to work with local, national and international communities to effect positive change in the world. Academic Community Engagement courses serve as cornerstones of the JPC program by preparing students to connect their academic experiences with public life and real world issues.