This document covers the required courses in computer science and engineering curriculum which were taught during the Spring 2010 semester.

The goal of this document is to identify the support the listed courses provide through associated student learning objectives for the ABET Engineering Accreditation Commission (EAC) and Computing Accreditation Commission (CAC) Criterion 3 Outcomes.

Following sections also provide the list of student learning objectives (SLO) associated with each course, and the support provided by these SLOs for a specific ABET EAC and CAC Criterion 3 outcome.

<table>
<thead>
<tr>
<th>Number</th>
<th>Course Name</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1560</td>
<td>Introduction to Object-Oriented Programming</td>
<td>Ledgard</td>
</tr>
<tr>
<td>1570</td>
<td>Linear Data Structures</td>
<td>Heuring</td>
</tr>
<tr>
<td>1580</td>
<td>Non-Linear Data Structures</td>
<td>Miller</td>
</tr>
<tr>
<td>1590</td>
<td>Discrete Structures</td>
<td>Ledgard</td>
</tr>
<tr>
<td>2000</td>
<td>EECS Professional Development</td>
<td>Heuring</td>
</tr>
<tr>
<td>2300</td>
<td>Electric Circuits</td>
<td>King</td>
</tr>
<tr>
<td>2550</td>
<td>Operating Systems and Systems Programming</td>
<td>Heuring</td>
</tr>
<tr>
<td>3200</td>
<td>Signals and Systems</td>
<td>Molyet</td>
</tr>
<tr>
<td>3400</td>
<td>Electronics I</td>
<td>Johnson</td>
</tr>
<tr>
<td>3550</td>
<td>Software Engineering</td>
<td>Ledgard</td>
</tr>
<tr>
<td>4000</td>
<td>Senior Design Project</td>
<td>Shenai</td>
</tr>
</tbody>
</table>

**EAC Criterion 3 Program Outcomes**

Engineering programs must demonstrate that their students attain the following outcomes:

a) An ability to apply knowledge of mathematics, science, and engineering

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
d) An ability to function on multidisciplinary teams

e) An ability to identify, formulate, and solve engineering problems

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 societal context

i) A recognition of the need for, and an ability to engage in life-long learning

j) A knowledge of contemporary issues

k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

EECS Course Support for ABET EAC Criterion 3 Outcomes is detailed in the following table:

<table>
<thead>
<tr>
<th>Number</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1560</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1570</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>1580</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1590</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2300</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2550</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>3400</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3550</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

CSE undergraduate degree program EAC Criterion 3 subcommittees are formed as below:

<table>
<thead>
<tr>
<th>ABET EAC Criterion 3 Outcomes</th>
<th>CSE Subcommittees</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, e</td>
<td>Miller (chair)</td>
</tr>
<tr>
<td></td>
<td>Alam</td>
</tr>
<tr>
<td></td>
<td>Nowlin (LCCC)</td>
</tr>
<tr>
<td>b, c, k</td>
<td>Kaur (chair)</td>
</tr>
<tr>
<td></td>
<td>Jamali</td>
</tr>
<tr>
<td></td>
<td>Miller</td>
</tr>
<tr>
<td></td>
<td>Kang (LCCC)</td>
</tr>
<tr>
<td>d, g</td>
<td>Standley (chair)</td>
</tr>
<tr>
<td></td>
<td>Ledgard</td>
</tr>
<tr>
<td></td>
<td>Jamali</td>
</tr>
<tr>
<td>i, j</td>
<td>Ledgard (chair)</td>
</tr>
<tr>
<td></td>
<td>Heuring</td>
</tr>
<tr>
<td></td>
<td>Kaur</td>
</tr>
</tbody>
</table>
CAC Criterion 3 Program Outcomes

The program has documented measurable outcomes that are based on the needs of the program’s constituencies.

The program enables students to achieve, by the time of graduation:

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
d) An ability to function effectively on teams to accomplish a common goal
e) An understanding of professional, ethical, legal, security and social issues and responsibilities
f) An ability to communicate effectively with a range of audiences
g) An ability to analyze the local and global impact of computing on individuals, organizations, and society
h) Recognition of the need for and an ability to engage in continuing professional development
i) An ability to use current techniques, skills, and tools necessary for computing practice.
j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choice.
k) An ability to apply design and development principles in the construction of software systems of varying complexity.

EECS Course support (through SLOs) for CAC Criterion 3 Outcomes is shown in the following table:
<table>
<thead>
<tr>
<th>Number</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1560</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>1570</strong></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1580</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1590</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2300</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2550</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3200</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3400</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3550</strong></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4000</strong></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The CSE undergraduate degree program faculty formed the following subcommittees to assess ABET CAC Criterion 3 outcomes a through k.

<table>
<thead>
<tr>
<th>ABET CAC Criterion 3 Outcomes</th>
<th>CSE Subcommittees</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, i</td>
<td>Miller (chair)</td>
</tr>
<tr>
<td></td>
<td>Alam</td>
</tr>
<tr>
<td></td>
<td>Nowlin (LCCC)</td>
</tr>
<tr>
<td>b, c</td>
<td>Kaur (chair)</td>
</tr>
<tr>
<td></td>
<td>Jamali</td>
</tr>
<tr>
<td></td>
<td>Miller</td>
</tr>
<tr>
<td></td>
<td>Kang (LCCC)</td>
</tr>
<tr>
<td>d, f</td>
<td>Standley (chair)</td>
</tr>
<tr>
<td></td>
<td>Ledgard</td>
</tr>
<tr>
<td></td>
<td>Jamali</td>
</tr>
<tr>
<td>e, g, h</td>
<td>Heuring (chair)</td>
</tr>
<tr>
<td></td>
<td>Standley</td>
</tr>
<tr>
<td></td>
<td>Jackson (LCCC)</td>
</tr>
<tr>
<td>j, k</td>
<td>Ledgard (chair)</td>
</tr>
<tr>
<td></td>
<td>Heuring</td>
</tr>
<tr>
<td></td>
<td>Kaur</td>
</tr>
</tbody>
</table>
The student will be able to...

1. Use Eclipse as an Integrated Development Environment (IDE).
2. Take a problem and develop a structure to represent objects and algorithms to perform operations.
3. Apply standards and principles to write truly readable code.
4. Test the program and, if necessary, find mistakes in the program and correct them.
5. Learn the fundamentals of input and output using the java.io library.
6. Design a class that serves as a program module or package.
7. Understand and demonstrate the concepts of object-oriented design, polymorphism, information hiding, and inheritance.
8. Develop applications using simple graphical user interfaces.
9. Become familiar with some of the common classes available in the Java language.

Outcome e
Supported by SLOs 2, 6, 7, and 8

Outcome k
Supported by SLOs 1 through 9

Outcome i
Supported by SLOs 1 through 9

Dr. Ledgard
STUDENT LEARNING OBJECTIVES

The student will be able to …

1. Develop a programming solution using stacks, linked lists, queues, and hash tables
2. Develop a solution using backtracking and understand the basic principles
3. Analyze simple program/algorithms and determine their true complexity
4. Give complexity bounds of common operations and data structures
5. Use at least two dimensional patterns to implement a programming solution
6. Implement sorting algorithms

EAC OUTCOMES SUPPORTED

Outcome a
Supported by SLO 3
Outcome e
Supported by SLOs 1, 2, and 6
Outcome k
Supported by SLO 1, 2, 5, and 6

CAC OUTCOMES SUPPORTED

INSTRUCTOR

Dr. Heuring
The student will be able to ...

1. Understand the C++ programming language, discuss the C++ primitive data types and built-in data structures, understand pointers, and dynamic memory, be able to compare and contrast the costs and benefits of dynamic and static data structure implementations, understand the C++ object-oriented concepts and principles, including encapsulation and information hiding, separation of behavior and implementation, classes and subclasses, inheritance, and polymorphism, and be able to use C++ in the construction of programs/software.

2. Understand and know how to build programs/software using one or more software development environments, including (but not limited to): MS Visual Studio/.Net, Eclipse, and GNU g++ with GDB.

3. Explain the use of big O, omega, and theta notation to describe the amount of work done by an algorithm, be able to use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms.

4. Describe the divide-and-conquer approach and know when a recursive solution is appropriate for a problem, and implement a divide-and-conquer algorithm to solve an appropriate problem.

5. Discuss the computational efficiency of the principal algorithms for sorting, discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application specific patterns in the input data, and implement the most common quadratic and $O(N \log N)$ sorting algorithms in a high-level language.

6. Understand the representations of graphs (adjacency list, adjacency matrix), graph traversal algorithms (depth-first search and breadth-first search), shortest path algorithms (Dijkstra’s and Bellman-Ford), and topological sort algorithms, be able to compare and contrast alternative implementations of them with respect to performance, costs, and benefits, and implement the user-defined data structures for graphs in a high-level language, and be able to write programs to solve problems using these fundamental graph algorithms.

7. Understand the representations of trees (binary trees, k-ary trees), binary search tree algorithms (basic binary search trees, red-black trees, AVL trees), and B-tree algorithms, be able to compare and contrast alternative implementations of them with respect to performance, costs, and benefits, be able to implement the user-defined data structures for trees in a high-level language, and be able to write programs to solve problems using these fundamental tree algorithms.

8. Evaluate and choose the appropriate data structure and its related algorithms for modeling a given problem, provide justification for that choice, and choose the appropriate implementation.
EAC OUTCOMES SUPPORTED

Outcome a  
Supported by SLOs 3, 4, 5, 6, 7 and 8
Outcome c  
Supported by SLOs 1 through 8
Outcome e  
Supported by SLOs 1, 3, 4, 5, 6, 7, and 8
Outcome k  
Support by SLOs 1 through 8

CAC OUTCOMES SUPPORTED

Outcome a  
Supported by SLOs 3, 4, 5, 6, 7, and 8
Outcome b  
Supported by SLOs 1, 3, 4, 5, 6, 7, and 8
Outcome c  
Supported by SLOs 1 through 8
Outcome i  
Supported by SLOs 1 through 8
Outcome j  
Supported by SLOs 3, 4, 5, 6, 7, and 8
Outcome k  
Supported by SLOs 1, 3, 4, 5, 6, 7, and 8

INSTRUCTOR

Dr. Miller
The student will be able to...

1. Learn and apply Mathematical Induction to a range of problems
2. Learn fundamental notations for sets, functions, sequences, and summations
3. Learn and apply Number Theory to solve problems in Computer Science
4. Develop the mathematical underpinnings required for the theory of computer security
5. Learn the elementary principles of Combinatorics to solve problems in Computer Science
6. Ability to recognize and use equivalence relations and partial orderings
7. Solve a limited class of recurrence relations
8. Ability to define and construct graphs and trees
9. Motivate the relevance of sound mathematics to software development

EAC OUTCOMES SUPPORTED

Outcome a
Supported by SLOs 1 through 9

CAC OUTCOMES SUPPORTED

Outcome a
Supported by SLOs 1 through 9

INSTRUCTOR

Dr. Ledgard
EECS 2000 PROFESSIONAL DEVELOPMENT

STUDENT LEARNING OBJECTIVES

The student will be able to ...

1. Understand the differences between patent, copyright, and trademark protection.
2. Understand modern issues in privacy – understand an issue and discuss as a paper.
3. Develop an appreciation of professional registration and life-long learning.
4. Write a business plan for a hypothetical startup company.
5. Discuss in class both expected behavior in a work environment as well as professional code of ethics.

EAC OUTCOMES SUPPORTED

Outcome f
Supported by SLO 1
Outcome g
Supported by SLO 4
Outcome i
Supported by SLO 3
Outcome j
Supported by SLOs 1 and 2

CAC OUTCOMES SUPPORTED

INSTRUCTOR

Dr. Heuring
The student will be able to . . .

1. Define voltage, current, energy and power in the context of an electric circuit
2. Define the terminal behavior of the basic linear circuit elements
3. Apply Kirchoff’s laws to the systematic formulation of a set of circuit equations
4. Apply Thevenin’s and Norton’s theorems to circuit analysis
5. Apply the voltage divider, current divider, superposition, and maximum-power transfer theorems to circuit analysis
6. Perform ac steady state circuit analysis using phasors
7. Calculate complex power in an ac system
8. Do transient analysis of first order circuits
9. Analyze a balanced 3-phase circuit
10. Recognize physical circuit elements in the lab and assemble a circuit from a schematic diagram
11. Experimentally measure voltage, current and power
12. Produce a written lab report in a standard format, which includes a brief discussion of relevant theory

---

**EAC OUTCOMES SUPPORTED**

Outcome a  
Supported by SLOs 1 through 9

Outcome e  
Supported by SLOs 10 and 11

Outcome g  
Supported by SLO 12

---

**CAC OUTCOMES SUPPORTED**

---

**INSTRUCTOR**

Dr. R. J. King
The student will be able to ...

1. Articulate the major areas of system software and programming.
2. Explain the objectives and functions of modern operating systems, analyze the tradeoffs inherent in operating system design, and identify some of the devices and resources an OS would manage.
3. Articulate the responsibilities and services provided by modern operating systems, and explain the range of requirements that an operating system has to address.
4. Articulate, compare and contrast, and know the data structures for supporting the primary O/S concepts of: interrupts and interrupt processing, concurrent processes, threads, synchronization, the memory hierarchy, memory management and virtual memory, scheduling, system state, user protection, and structuring of an operating system.
5. Utilize common programming tools and environments in the implementation of system level software.
6. Code algorithms that interact with “lower level” system details and the operating system interface.
7. Design and implement program solutions for larger and more detailed projects.

EAC OUTCOMES SUPPORTED

Outcome a
Supported by SLOs 1, 2, 3, 4, 6, and 7

Outcome c
Supported by SLOs 6 and 7

Outcome k
Supported by SLOs 4, 5, 6, and 7

CAC OUTCOMES SUPPORTED

INSTRUCTOR

Dr. Heuring
STUDENT LEARNING OBJECTIVES

The student will be able to ...

1. The student will be able to represent and classify signals and systems.
2. The student will be able to represent and apply singularity functions.
3. The student will be able to obtain the response of a continuous, linear, time-invariant, causal system by using convolution.
4. The student will be able to obtain the Fourier series expansion of a periodic signal and apply it to continuous, linear, time-invariant systems.
5. The student will be able to obtain and plot the Fourier transform for simple aperiodic continuous-time signals.
6. The student will be able to utilize the Laplace transform method to solve continuous, linear, time-invariant systems and to obtain transfer functions.
7. The student will be able to analyze continuous, linear time-invariant systems using state variable formulation and solve the resulting state equations.
8. The student will be able to convert a continuous-time signal to the discrete-time domain and reconstruct it using the sampling theorem.
9. The student will be able to utilize the z-transform method to solve linear discrete-time systems and to obtain transfer functions.
10. The student will be able to use MATLAB and Simulink to solve continuous and discrete-time systems.

EAC OUTCOMES SUPPORTED

Outcome a
Supported by SLOs 1 through 9
Outcome k
Supported by SLO 10

CAC OUTCOMES SUPPORTED

INSTRUCTOR

Dr. Richard Molyet
STUDENT LEARNING OBJECTIVES

The student will be able to ...

1. Understand the qualitative and quantitative properties of semiconductor materials that are used in fabrication of electronic circuit components.
2. Understand the mechanisms of motion of electric charges in semiconductor materials.
3. Learn the concepts of large and small signals, and the application areas of corresponding electronic component models.
4. Gain the present day understanding level of the processes which govern the behavior of pn-junctions.
5. Apply the large signal method of analysis to nonlinear electric circuits (and systems).
6. Learn the properties and operation modes of the four types of FETs.
7. Apply the large signal method of analysis to electronic circuits that contain FETs.
8. Apply the SPICE simulation method of analysis to electronic circuits that contain FETs.
9. Apply the approximate large signal method of analysis to electronic circuits that contain BJTs.
10. Design BJT inverter circuits with a required noise margin and fan-out.
11. Learn how to design the BJT inverter circuit: (a) of minimum size, (b) with equal rise and fall times, and (c) that has a required logic threshold voltage value.
12. Learn the ways of lowering power dissipation in digital electronic circuits.
13. Learn the trade-off between power dissipation and time delay of digital electronic circuits.
14. Analyze static (and the dynamic) combinational logic circuits with the goal of determining the Boolean function implemented by the circuit.
15. Design circuits of combinational static CMOS gates, so that they implement a desired Boolean function.
16. Understand the influence of adjusting transistor aspect ratios on the dynamic performance of logic gates.
17. Design the transistor aspect ratios of a given static CMOS gate, so that the gate has the same rise and fall times as the reference inverter.
18. Understand the three established principles of encoding the logic/numeric values in memory cells: state of a bistable circuit, electrical charge on a capacitance, and a FET’s threshold voltage value.
19. Understand the basic technological challenge to each of the three encoding principles: power dissipation, leakage of electrical charge, extra high voltages involved in changing the threshold voltage of FETs.
20. Understand the complexity of modern memory arrays and their design challenges.
21. Work with the TTL family legacy circuits,
22. Understand the merit of ECL circuits in today’s multi GHz communication systems.
23. Learn the basic principle of D/A conversion,
24. Learn one simple A/D conversion approach.
25. Complete the lab assignment on D/A and A/D conversion.

EAC OUTCOMES SUPPORTED

 Outcome b
 Supported by SLOs 23, 24, and 25
 Outcome c
 Supported by SLOs 10, 11, 15, and 17
 Outcome j
 Supported by SLO 12 and 20
 Outcome k
 Supported by SLOs 5, 7, 8, and 9

CAC OUTCOMES SUPPORTED


INSTRUCTOR

Dr. Johnson
The student will be able to...

1. Learn the canonical metaphor for building large software projects
2. Review the many other models of developing software projects
3. Learn the skills required to become a true software craftsman
4. Learn the roles and skills required for working as a team on a software project
5. Work in a team to build a software product
6. Recognize and create a Functional Specification from a set of User Requirements
7. Be able to define the properties of readable and reusable code
8. Conduct a specification, design, or code review
9. Make an effective oral presentation on a technical topic
10. Be knowledgeable of contemporary issues related to software
11. Identify some of the software issues that affect society as a whole
12. Plan and execute lifecycle steps for developing a complex software product

EAC OUTCOMES SUPPORTED

Outcome c
Supported by SLOs 1, 2, 3, 4, 5, 6, 7, 8, and 12

Outcome d
Supported by SLOs 3, 4, 5, and 8

Outcome g
Supported by SLOs 8 and 9

Outcome h
Supported by SLOs 10 and 11

Outcome j
Supported by SLOs 2, 10, and 11

Outcome k
Supported by SLOs 1 through 12

CAC OUTCOMES SUPPORTED

Outcome c
Supported by SLOs 1 through 12

Outcome d
Supported by SLOs 3, 4, 5, and 8

Outcome f
Supported by SLOs 8 and 9

Outcome g
Supported by SLOs 10 and 11

Outcome i
Supported by SLOs 1, 2, 3, 4, 5, 6, 7, 8, and 12

Outcome k
Supported by SLOs 1 through 12
INSTRUCTOR

Dr. Ledgard
The student will be able to...

1. Design a complex system (or component or process) to realistic performance specifications in compliance with applicable engineering standards and multiple realistic constraints, and report the results through a comprehensive and professional technical write-up and oral/poster presentation.
2. Propose a solution or critique a proposed solution to an engineering problem, identifying possible negative regional, national, global or societal consequences and recommending ways to minimize or avoid them.
3. Find relevant sources of information about a specified topic in the library and on the World Wide Web (or perform a full literature, patent, and product search).
4. Function effectively on a team project, with individual effectiveness being determined by instructor observation, peer ratings, and self-assessment.
5. Build a prototype of a design and demonstrate that it meets performance specifications.
6. List and discuss several possible reasons for deviations between predicted and measured results from an experiment or design, and choose the most likely reason and justify the choice.
7. Given a job-related scenario that requires a decision with ethical implication, the student will be able to identify possible courses of action and discuss the pros and cons of each one, pick the best course of action and justify the decision.
8. Write an effective technical correspondence (i.e. abstract, executive summary, project report) or give an effective oral presentation.

EAC OUTCOMES SUPPORTED

Outcome b
Supported by SLOs 4 through 6
Outcome c
Supported by SLOs 1 through 3
Outcome d
Supported by SLOs 4 through 6
Outcome f
Supported by SLO 7
Outcome g
Supported by SLO 8
Outcome j
Supported by SLOs 1 through 3
CAC OUTCOMES SUPPORTED

Outcome c
Supported by SLOs 1, 5, 6, and 9
Outcome d
Supported by SLO 4
Outcome e
Supported by SLOs 2 through 7
Outcome f
Supported by SLOs 4 and 8
Outcome h
Supported by SLO 3
Outcome i
Supported by SLO 5

INSTRUCTOR

Dr. Shenai