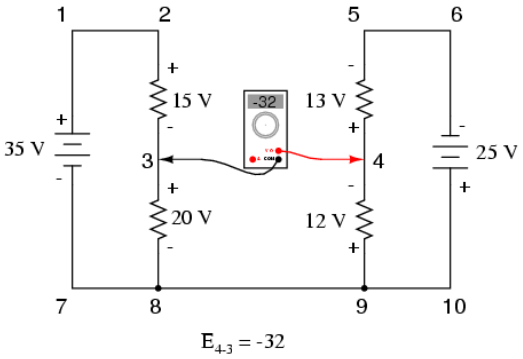
EELE 101 Introduction to Electrical Fundamentals

Dept. of Electrical and Computer Engineering - Montana State University
Spring 2016 Course Syllabus (v.1)



Course Description:

This course provides a hands-on introduction to a number of different areas in Electrical and Computer Engineering, the applications of these technologies to solve real-world problems, and the potential impacts on society in general. It incorporates lectures, laboratory experiences, and programming exercises that introduce you to the fundamentals of electrical and computer engineering. Topics include Kirchhoff's and Ohm's Laws, using meters and oscilloscopes, time-varying signals in electric circuits, resistors, capacitors, series and parallel circuits, introduction to digital circuits, introduction to programming, problem solving including computer applications, technical communications, and team work. The course culminates in a final project to build and program an autonomous car to complete a challenge (a line-following race course).

Instructor:
Prof. David Dickensheets
Office: 530 Cobleigh Hall
Phone: 994-7874
email: davidd@montana.edu

TAs:
Trevor Vannoy, Office: 510 Cobleigh Hall, email: trevor.vannoy@msu.montana.edu
Krishna Chattergoon, Office: 511 Cobleigh Hall, email: krish_chatter@live.com
Skylar Tamke, Office: 510 Cogleigh Hall, email: skylar4494@gmail.com

Prerequisite: Math 151 Precalculus (was Math 160)

D2L Tab: Intro Electrical Fundamentals CL - EELE_101_001t005_201630_CL
We will be using the Brightspace (Desire2Learn) learning management system for this course. The address is: https://ecat1.montana.edu/ . You will log into this system using your MSU ID and password (the same as your MyInfo login). All course materials can be found on this site in addition to your current grades and the most up-to-date course schedule. If you have technical difficulties with the system (i.e., can't login, the system is down, etc…), contact the MSU IT Center help desk at 994-3255 (D2LL).

Text: *EE 101 Course Note Set*, available as PDF files on D2L under "Course Notes"

Course Components
This is a 3-credit course that contains 3 different components: a 1 credit lecture, a 1 credit laboratory, and 1 credit of recitation. You must register for a section of each component.

Lecture: The lecture component of this course will cover the fundamentals of electrical quantities, circuit elements, circuit laws, signal representation, and math skills. The lecture will also discuss the electrical/ computer engineering profession, career opportunities, engineering research, and the role of electrical/computer engineers in society. The lecture portion of the course will meet once per week for 50 minutes. There will be weekly homework assignments that must be completed (paper and pencil) and turned in for grading.

Laboratory: The laboratory portion of this course will provide hands-on experience with building circuits, measuring electrical quantities, and debugging electrical systems. The first half of the semester will focus on building circuits using a solderless breadboard. The second half of the semester will focus on building and programming the NXP (was Freescale) Cup Formula Car. The laboratory will meet once per week for 2 hours in Cobleigh 602. There will be laboratory write-ups that must be completed and turned in each week.

Recitation: The recitation portion of this course will present an introduction to the C programming language. During the first half of the semester, the recitation will cover programming using a traditional development environment using the console of a computer for input/output. During the second half of the semester the recitation will cover how to program the NXP Cup Formula Car. The recitation is designed for a flipped learning experience in which the C programming lectures are provided as videos that are to be completed prior to the recitation period. Students are then expected to complete an on-line quiz and upload an attempt at their programming assignment to the D2L Dropbox prior to coming to the recitation session. During the recitation session, students will demonstrate their functioning program or receive help with their assignment. A sign-off sheet is provided for each programming assignment that must be signed by the instructor for credit. Students are then required to upload their final program code to the D2L dropbox to complete the assignment. The recitation will meet once per week for 1 hour in Cobleigh 602.

The course will culminate with an NXP Cup Formula Car race competition. Students will program their cars to follow a line course and compete for the fastest race time. The race will take place during the engineering design fair the week before final exams.

Lab Supplies: You will be expected to purchase a solderless breadboard (approximately $24.00) and a Freescale Semiconductor Formula Car Development Kit (approximately $160 for engineering majors, $210 for non-majors; the difference comes from Engineering Program Fees). These may be purchased from the ECE Stockroom, 622 Cobleigh Hall. You will need the breadboard by the first lab session. You will need the Freescale Cup car kit by Week 6 since you will be targeting the Freescale microcontroller board that comes with the car kit.

Grading Scheme:

Homework	20%	-- Includes any in-class and take home assignments
Labs	35%	-- Weekly assignments due at the beginning of the follow lab
C Programming Quizzes	5%	-- Online quizzes based on on-line tutorials
Pre-Recitation Code Upload	5%	-- An attempt at the assignment, turned into D2L Dropbox
C Programming Demos	10%	-- Demonstration of working code done in recitations
Post-Recitation Code Upload	5%	-- Well commented code, turned into D2L Dropbox
Midterm Exam	10%	-- Scheduled March 26, during the lecture period
Formula Races and Final Exam	10%	-- To be scheduled near or during Finals week

Attendance/Absence Policy:
While attendance will not be taken during lecture sessions, it is important that you participate in each lecture as a good portion of the course material will be delivered solely through the lecture. Attendance will be taken at each laboratory and recitation session. Any non-emergency absence from a laboratory session must be excused prior to the absence. In the case of an emergency absence, documentation of the emergency will be required to allow for making up the lab work.

When is work due?
There are a number of different types of assignments associated with this class. A calendar showing the plan for the course is available on the D2L page. Each individual assignment will be discussed in lecture and/or be available for download from the D2L page. In general, course assignments are due as follows:

- **Homework assignments** are due at the beginning of lecture. You have an assignment due most (but not all) weeks. The homework assignments can be found on the D2L page. These are paper assignments, handed in during lecture.
- **Lab activity write-ups** are typically collected at the end of the lab period, but are due by 5 PM on Friday the week of the lab. Please download, print out, and carefully read each lab assignment BEFORE you come to lab each week. These are paper assignments collected in lab by the TA.
- **C Programming Quizzes** are **to be completed on-line** before the recitation section each week. You should watch the "on-line C programming lecture" in D2L before attempting the quiz.
- **Pre-Recitation C Programming Code** is to be uploaded to the D2L dropbox by midnight the day before your recitation each week. This attempt does not need to work, but does need to show that you have read the assignment and at least have made an attempt.
- **C Programming Demos** are to be completed during the recitation session, including review and check-off by the course instructor or TA. This is a paper checkoff form.
- **Post-Recitation C Programming Code** In addition to demonstrating your code with the TA watching, you must submit **fully documented, operational code** to the D2L dropbox by midnight Monday following the recitation section.

Late Work Policy
The due date for each assignment is described above. Late assignments will be accepted up to one week after the due date *with a 50% late penalty*. No assignments will be accepted for grading more than one week after the original due date.

Collaboration Policy
In general, working with other students in this course is encouraged. However, all assignments turned in for grading must: (1) be wholly and completely the work of the author (2) reflect each individual student's understanding of the course material For example, you will work with a partner on the laboratory exercises. You are encouraged to work together to figure out any challenging aspects of the exercise, and help each other debug your circuits. Each student should then individually prepare her/his own lab write-up. You and your partner might both have the same experimental data in your reports; however, your analysis and interpretation of these results should NOT be identical! In other words, you may work with others to improve your understanding of course concepts, but by the time you turn in an assignment for grading, you should be able to do the entire assignment by yourself.

Cheating (or any other form of academic misconduct) on any graded component of this course will not be tolerated, and will be penalized in accordance with MSU policy. In general, this course will follow the policies described in the Conduct Guidelines and Grievance Procedures for Students (http://www2.montana.edu/policy/student_conduct/student_conduct_code.htm).

Personal electronics policy
To avoid distracting your classmates and your instructors, please turn off (or at least put into silent mode) any cell phones, PDAs, or other personal electronic devices. If you absolutely must take a call, send/receive a text or otherwise use a personal electronic device, please excuse yourself from the classroom or laboratory.

General Comments:
EELE 101 should be a fun and challenging course in which you begin to learn the basics of electrical engineering. You will be required to develop and to use math skills, take key electrical measurements using electronic test

equipment, learn basic C programming and receive an introduction to the industry-standard engineering computation platform, MATLAB.

The goal of any engineering curriculum should not only be to teach you a set of practical and important information and skills, but it should help you develop your ability to think critically and to reason. A list of the goals for this course includes:

• **Fundamental Electrical Quantities and Circuit Elements**: You should be able to:
  (1) Recall the meaning and the units of measure for charge, current, electrical potential, and power,
  (2) Identify the circuit symbols for voltage and current sources, resistors, capacitors and diodes,
  (3) Describe the electrical properties of the resistors, capacitors and diodes,

• **Circuit Laws**: You should be able to:
  (1) Recall the definitions of Ohm's Law, Kirchhoff's Voltage Law and Kirchhoff's Current Law,
  (2) Be able to apply the circuit laws to find voltages and currents in multi-resistor, multi-source circuits

• **Waveforms and Math Skills**: You should be able to:
  (1) Describe the properties of DC and AC signals,
  (2) Identify the amplitude, frequency, and phase shift with respect to a reference given an equation of a waveform or a graphical representation of a waveform,
  (3) Write an equation describing a waveform given a graphical representation,
  (4) Manipulate complex numbers in rectangular and polar form.

• **Lab Skills**: You should be able to:
  (1) Understand the proper operation and be able to use a bench-top power supply,
  (2) Understand the proper operation and be able to use a digital multimeter,
  (3) Understand the proper operation and be able to use a function generator,
  (4) Understand the proper operation and be able to use an oscilloscope,
  (5) Be able to construct circuits using a protoboard (breadboard),
  (6) Understand the proper operation and be able to use a soldering iron in assembling electric circuits,
  (7) Be able to use MATLAB for the purposes of EE 101

• **C Programming:** You should be able to:
  (1) Understand the basic data types in C,
  (2) Understand how to use basic operators in C,
  (3) Understand how to control program flow,
  (4) Understand how functions work,
  (5) Be able to write simple programs in C