

Summer 2005
Introduction to Computers and Programming
Tentative Syllabus

People:

Paul J. Kominsky paulko@umich.edu www-personal.engin.umich.edu/~paulko/e101s
John W. Norton, Jr. jnorton@umich.edu www.johntheengineer.com/ENG101/ENG101.htm

Location:

Lectures are in Chrysler auditorium. Labs are in rooms 3rd floor Duderstadt or 2230 SRB.

Purpose of course:

“Educational Objectives” from the official course description: The educational objectives of Engineering 101, Introduction to Computers and Programming, center on students learning to formulate and implement algorithms to solve problems in math, science and engineering.

Students who pass the class will ably demonstrate the following abilities:

Define an algorithm, Describe standard encodings of data, Describe the place value system, Write a positive integer in binary, decimal and hexadecimal, Write a negative integer using twos-complement, Describe the IEEE standard floating point encoding, Describe data types and identifiers in C++ and Matlab, Classify operators as arithmetic, relational, logical or assignment, Identify statements as declarations, expressions, selection or iteration, Describe the role of sequence, selection and iteration in algorithms, Describe functions/procedures in a programming language, Define pass-by-value and pass-by-reference semantics, Describe files and i/o streams, Compute the number of bits required to encode data with a finite range, Use compiler/interpreter error messages to find a syntax error in a C++ or Matlab code, Recognize that a problem will require a selection statement, Recognize that a problem will require iteration or looping, Distinguish a counting loop from a more general iteration, Recognize the "input" and "output" parameters to a function, Trace C++ or Matlab code and determine the values of variables at each statement, Write code that requires selection/conditional execution, Write iteration loops controlled by a "while" condition, Write counting loops, including loops to accomplish array traversals, Write functions using call-by-value semantics, Write functions using call-by-reference semantics, Implement a given algorithm in C++ or Matlab, Debug a C++ or Matlab code to find a semantic or logic error, Create an algorithm to solve a physical or mathematical problem, Decompose a complex problem into simpler tasks, Devise functions/procedures that appropriately encapsulate a task.

....and that's all! Heh. Actually, most of the “real” ENG101 courses don't cover all of these subjects in depth, and many of these abilities are fairly straightforward within the context of this course.

THIS course, the summer course all of you are attending, is designed to boost your chances of success in the “real” ENG101 course when you take it this coming Fall or Winter. There are two

aspects of doing well, learning the material, and learning HOW to learn the material. Most of you are expects at learning material. However, it is likely that few of you have ever thought of how your learned new material, or how to go about maximizing your learning potential.

This summer course will push you fast, and you will not believe that you will actually be pushed that fast this fall. You will be right. You will actually be pushed even harder this fall! New information will come at you like water out of a hose, drowning you. You will scramble to keep up with the flood, projects will be delayed and postponed, and other frustrations will occur. However, through it all, realize **that you have the ability and talent to survive** (and do well!) at this institution.

Summer course typical week:

Time	Monday	Tuesday	Wednesday	Thursday
6:30	Lecture	Lecture	Open lab	Lecture
7:30	Lab	Lab	Open lab	Office hours

Calendar:

	Lecture	Lab	Assignment
June 27 – Monday	What is science? Try things! (Paul)	Binary math	
1 June 28 – Tuesday	html programming (Paul)	making a webpage (Paul)	HW 1. Personal webpage, main page with link to resume
June 30 – Thursday	INITIAL ASSESSMENT EXAM!		
July 4 – Monday	- - Forth of July break - -		
2 July 5 – Tuesday	Flow charts, branching – pancake example – flour and sugar (John)	Linux/AFS/Hello World	HW 2. Making change
July 7 – Thursday	Binary math – LAB!	Office hours	HW 1 finished.
July 11 – Monday	Flow charts, branching, if/while (John)	Al kashi algorithm example	
3 July 12 – Tuesday	For loops, min/max –limitations of current variables – motivate VECTORS! (Paul)	Examples of different types of loops, nested loops	
July 14 – Thursday	Vectors – introduction, Game of life rules (John)	Office hours	HW 3. Game of life

	Lecture	Lab	Assignment	
	July 18 – Monday	Vector example, bubble sort (John)	Using vectors in code, basic vector commands	HW 2 finished.
4	July 19 – Tuesday	Iterating a matrix Printing vectors – motivate FUNCTIONS! (Paul)	Show Matlab Life example	
	July 21 – Thursday	Functions, pass-by-value Functions, pass-by-reference (John)	Office hours	
5	July 25 – Monday	Vectors of vectors, matrices (John)	Example boundary checking	
	July 26 – Tuesday	Using matlab (Paul)	Matlab lab examples	
	July 28 – Thursday	QUIZ!	Office hours	HW3 finished.
6	August 1 – Monday	Random numbers, seeds (John)	Using random numbers, Monte carlo method	HW 4. Blackjack (except PTP)
	August 2 – Tuesday	Rules of Blackjack (Paul)	Work on blackjack assignment	
	August 4 – Thursday	Monte Carlo method (Paul)	Office hours	PTP Gone! Bye! HW 4 part1 finished.
7	August 8 – Monday	Comparison of functions, passing variable types, passing vectors (John)	Defining a function	
	August 9 – Tuesday	Scope! (Paul)	Functions	
	August 11 – Thursday	Review, help with HW 5 (Both)	Office hours	HW4 part 2 finished.
8	August 15 – Monday	FINAL ASSESSMENT EXAM!		
	August 16 – Tuesday	In class review		
	August 17 – Wednesday		*** PACK ***	
	August 18 – Thursday		*** GO HOME!! ***	