COURSE OUTLINE – subject to changes :

L-1	Mon 16 Jun	welcome to mcs 260 – computer literacy and Python programming computer architecture – first steps with Python
L-2	Wed 18 Jun	the von Neumann machine – calculating in the Python shell numbers, variables, and assignments – developing Python programs
L-3	Fri 20 Jun	operating systems – the os module – turtle graphics syntax and semantics of languages – strings, lists, and tuples
L-4	Mon 23 Jun	mass storage, files and databases – dictionaries in Python – anydbm boolean algebra – flowcharts – conditional constructs in Python
L-5	Wed 25 Jun	transistors and logic gates – intrinsic operations on numbers and strings flip-flops and registers – intrinsic operations on lists: queues and stacks
Project One due on Friday 27 June at noon		
L-6	Fri 27 Jun	adder circuits – loop constructs: the while and for – estimating π simulation using random numbers – binary expansion with repeat until: break
L-7	Mon 30 Jun	top down design of programs – functions in Python local and global variables – arguments of functions – functions using functions
L-8	Wed 2 Jul	functional programming – lambda forms – list comprehensions organization of data on files – manipulating files with Python
	Fri 4 Jul	**Independence Day holiday. No Classes.**
L-9	Mon 7 Jul	data compression – format conversions – using buffers to process files networking and the internet – markup languages – using the urllib module
Project Two due on Wednesday 9 July at noon		
L-10	Wed 9 Jul	review for the midterm exam
L-11	Fri 11 Jul	**midterm exam**
L-12	Mon 14 Jul	software engineering – bottom up design of programs – modules in Python the software development cycle and quality – modular design with Python
L-13	Wed 16 Jul	software licensing and open source – modules and packages – NumPy/SciPy object-oriented programming – unified modeling language – classes in Python
L-14	Fri 18 Jul	data and functional attributes of classes – operator overloading encapsulation, inheritance, polymorphism – wrapping and delegation
L-15	Mon 21 Jul	software testing, verification techniques – pre- and postconditions with assert exception handling – defining, raising, and handling exceptions in Python
Project Three due on Wednesday 23 July at noon		
L-16	Wed 23 Jul	complexity and cost – complexity classes – timing Python programs graphical user interfaces – using Tkinter, a GUI toolkit for Python
L-17	Fri 25 Jul	expression evaluation – graphing functions – using the Canvas widget entering data with a scale – the development of animations
L-18	Mon 28 Jul	class hierarchies – visualization with inheritance cellular automata – the game of life – a Tkinter application
L-19	Wed 30 Jul	processes and threads – lifecycle of a thread – multithreaded programming an object-oriented implementation of a predator-prey simulation
L-20	Fri 1 Aug	network programming with sockets – the multiprocessing module high level parallel programming – the multiprocessing module
Project Four due on Monday 4 August at noon		
L-21	Mon 4 Aug	recursive algorithms – tracing the execution of recursive functions recursive images – drawing fractals with Tkinter
L-22	Wed 6 Aug	review for the final exam
L-23	Fri 8 Aug	**final exam**