



CSC 112: An Introduction to Programming Using Python

Eric Patterson, Ph.D.

(910) 962-7701

pattersone@uncw.edu

Fall 2009 Schedule

Week 0	August		19	20	21
Course Introduction					
Week 1		24	25	26	28
Computers, Languages, and Python: Ch 1, 2					
Last Drop/Add					
Week 2	September	31	1	2	3
Commands, Conditionals, and Loops: Ch 2, 3					
Week 3		7	8	9	11
Functions: Ch 3					
Labor Day					
Week 4		14	15	16	18
Basic Data Structures: Ch 4, 5					
Week 5		21	22	23	25
File Handling: Ch 6					
Week 6	October	28	29	30	1
Classes: Ch 7					
Week 7		5	6	7	9
Functional Design: Ch 8					
<-- Fall Break -->					
Week 8		12	13	14	16
Object-Oriented Design: Ch 9					
Week 9		19	20	21	23
Scopes, Name Spaces, and Modules: Ch 10					
Week 10		26	27	28	30
Some Advanced Python Features: Ch 11					
Week 11	November	2	3	4	6
Application Week 0: GUI Programming					
Week 12		9	10	11	13
Application Week 1: Web Apps					
Week 13		16	17	18	20
Application Week 2: Graphics					
Week 14		23	24	25	27
Application Week 3: Gaming					
26 <--Thanksgiving-->					
Week 15	December	30	1	2	
Wrap-up					

Required Text:

Exploring Python by

Timothy A. Budd

Beautiful is better than ugly.
 Explicit is better than implicit.
 Simple is better than complex.
 Complex is better than complicated.
 Flat is better than nested.
 Sparse is better than dense.
 Readability counts.
 Special cases aren't special enough to break the rules.
 Although practicality beats purity.
 Errors should never pass silently.
 Unless explicitly silenced.
 In the face of ambiguity, refuse the temptation to guess.
 There should be one -- and preferably only one -- obvious way to do it.
 Although that way may not be obvious at first unless you're Dutch.
 Now is better than never.
 Although never is often better than *right* now.
 If the implementation is hard to explain, it's a bad idea.
 If the implementation is easy to explain, it may be a good idea.
 Namespaces are one honking great idea -- let's do more of those!

-- Tim Peters