

BRONX COMMUNITY COLLEGE
of the City University of New York
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

SYLLABUS: CSI32 Introduction to Computer Programming II 3 credits/4 hours
(Object-Oriented Programming)

PREREQUISITE: CSI31 or department permission, ENG02, RDL02 if required.

TEXT: Object-Oriented Programming in Python, by Goldwasser and Letscher,
Pearson/Prentice-Hall, 1st Edition, 2008

Online reference for UML: <http://www.omg.org/docs/formal/03-03-01.pdf>

Software: Python v2.3 or later; Dia v0.96 (recommended structured diagram software)

Content:

Basics of object-oriented design (OOD) and object-oriented programming (OOP) using the Python language. Unified Modeling Language (UML) diagrams (class, sequence, activity and state diagrams) as design and programming tools. Modules, types, classes, inheritance, composition, methods, constructors. Recursion (structural and procedural).

Objectives:

1. To deepen the student's understanding of Python as an OO language to a level where other OO languages such as C++ or Java can be easily assimilated.
2. To regard every variable as an object of some class, and to review the built-in types from this perspective.
3. To provide the student with opportunities to use OOD/OOP to design correctly and to implement web applications and event-driven GUI applications.

Topic	Section	Assignments	Projects
Data and Types; Functions and Algorithms (UML activity diagram, i.e. flowchart)	1.1-1.2	p.29:1.5,1.6,1.10	
Higher Level Languages; Objects and Classes: OO Design (UML class diagram, UML sequence diagram)	1.3-1.5	p.29:1.15,1.19, 1.22,1.25,1.29,1.31	
Built-in Python classes (list, str) and numeric types (int, long, float)	2.2-2.5	p.82-4:2.5, 2.8, 2.9, 2.14, 2.18, 2.24(a-g)	p. 86:2.33
Expressions, Calling Functions	2.6-2.8	p.84:2.24(j-r),2.25, 2.27(a-r)	
Lists and for loops	4.1,4.5	p.151-6:4.5, 4.9, 4.13, 4.34	p. 156:4.39
While loops, defining functions (review)	5.1, 5.4	p.195-9:5.4, 5.5, 5.23	

Conditional statements (review)	4.4	p.154-5::4.23, 4.27	
Designing and implementing classes---a Fraction class	6.4	p.233:6.10	p.233:6.18 (or complex numbers)
Error checking and exceptions	5.5	p.200:5.33,5.34	
Design and Documentation	7.2, 7.4-7.6	p.268:7.4,7.6,7.8	
Modules and Unit Testing	7.7	p.269:7.6,7.8	
Input and Output; Files	8.1-8.5	p. 293:8.5, 8.13	p. 296:8.21-8.24
Graphics	3.1-3.3	p.121:3.5	p. 123: 3.17
Inheritance	9.1-9.4	p.328:9.3,9.6	
Structural recursion---Drawing a pyramid	4.3,11.1	p.151:4.7,4.10 p.390:11.4	
Procedural recursion---Binary search	11.3, 11.4	p.391:11.6 p.395:11.31	
Container Classes (list vs. tuple; Dictionary)	12.1-12.2	p.433:12.5	p.434:12.14
Event-driven programming (UML state diagram)	15.1	p.519:15.1,15.2	
Event-handling	15.2-15.4	p.520:15.6	p. 520:15.12 (or other GUI Project)
A Network Primer, Basic Client	16.1-16.2	(handout assignment: modify fig. 16.5)	
Basic Server	16.3	p.556:16.1	
A Chat Room	16.4		One of p. 557: 16.8-16.12

GJL 10/08