

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

**SYLLABUS: CSI32 Introduction to Computer Programming II      3 credits / 4 hours**

**PREREQUISITE: CSI31 or departmental permission, ENG02, RDL02 if required**

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

**Software: Python 2.6; DIA v0.97 (recommended structured diagram software)**

**Content:**

In this class we will study basics of object-oriented design (OOD) and object-oriented programming (OOP) using the Python language. Learn to use Unified Modeling Language (UML) diagrams (class, sequence, activity and state diagrams) as a design tool. Discuss modules, types, classes, inheritance, methods, constructors, and recursion.

**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 a programming project.

Students will complete 8 to 10 small programming assignments selected from the list of suggested exercises or comparable assignments developed by the instructor, and one bigger (final) project.

<b>Sections and Topics</b>	
<b>Chapter 1 Cornerstones of Computing (2 classes)</b>	
1.1 Data and Types	1.4 The Object-Oriented Paradigm
1.2 Operations, Functions, and Algorithms	1.5 Design and Modeling
1.3 High-Level Programming Languages	1.6 Chapter Review
<b>Suggested Review Questions</b> 1.6.3 Exercises	pp. 28 – 30 / 1.1 – 1.32 (practices and exercises)
<b>Chapter 2 Getting Started in Python (4 classes)</b>	
2.2 Using objects: the <b>list</b> Class	2.6 Calling Functions
2.3 Other Sequence Classes: <b>str</b> and <b>tuple</b>	2.7 Python Modules
2.4 Numeric Types: <b>int</b> , <b>long</b> , and <b>float</b>	2.8 Expressions
2.5 Type Conversions	
<b>Suggested Review Questions</b> 2.11.3 Exercises	pp. 82 – 85 / 2.1 – 2.21 (practices and exercises)

<b>Chapter 3 Getting Started with Graphics (2 classes)</b>	
3.1 The Canvas	3.3 Rotating, Scaling, and Flipping
3.2 Drawable Objects	3.4 Cloning (optional)
<b>Suggested Review Questions</b> 3.10.3 Exercises	pp. 121 – 122 / 3.1 – 3.6
<b>Chapter 4 (2 classes)</b>	
4.1 For loops (review)	4.4 Conditional Statements (review)
4.2 Case Study: DNA to RNA Transcription	4.5 List Comprehension
4.3 Case Study: Drawing a Pyramid (will be covered in Chapter 11)	4.6 Chapter Review
<b>Suggested Review Questions</b> 4.6.3 Exercises	pp. 150 – 157 / 4.1 – 4.3, 4.18 – 4.39
<b>Chapter 5 Additional Control Structures ( 2 classes)</b>	
5.1 While loops (review)	5.5 Error Checking and Exceptions
5.4 Case Study: Computing the Square Root	5.7 Chapter Review
<b>Suggested Review Questions</b> 5.7.3 Exercises	pp. 195 – 201 / 5.1 – 5.37
<b>Chapter 6 Defining Our Own Classes (2 classes)</b>	
6.4 A Fraction Class	
<b>Suggested Review Questions</b> 6.6.3 Exercises	p. 234 / 6.15, 6.18
<b>Chapter 8 Input, Output and Files (2 classes)</b>	
8.1 Standard Input and Output	8.4 Handling Newline Characters
8.2 Formatted Strings	8.5 Case Studies
8.3 Working with Files	8.6 Chapter Review
<b>Suggested Review Questions</b> 8.6.3 Exercises	pp. 293 – 297 / 8.1 – 8.24

<b>Chapter 9 Inheritance (2 classes)</b>	
9.1 Augmentation	9.3 When Should Inheritance (Not) Be Used
9.2 Specialization	9.4 Class Hierarchies and cs1graphics
<b>Suggested Review Questions</b> 9.7.3 Exercises	pp. 327 – 330 / 9.1 – 9.9
<b>Chapter 11 Recursion (3 classes)</b>	
11.1 A Bullseye Class (and section 4.3 is covered here)	11.4 Binary Search
11.3 Functional Recursion	
<b>Suggested Review Questions</b> 11.6.3 Exercises	pp. 390 – 395 / 11.1 – 11.6, 11.19 – 11.31
<b>Chapter 12 More Python Containers (2 classes)</b>	
12.1 Two Familiar Containers: <b>list</b> and <b>tuple</b>	12.2 Dictionaries
<b>Suggested Review Questions</b> 12.8.3 Exercises	pp. 433 – 434 / 12.1 – 12.6
<b>Chapter 15 Event-Driven Programming (3 classes)</b>	
15.1 Basics of Event-Driven Programming	15.4 Programming Using Events
15.2 Event Handling in our Graphics Module	15.6 Chapter Review
15.3 The Event Class	
<b>Suggested Review Questions</b> 15.6.3 Exercises	pp. 519 – 520 / 15.1 – 15.8, 15.9 – 15.12