Description of Course:
This course offers a general introduction to Computer Science and the fundamental principles of computer programming using the Python programming language. The course will introduce students to the process of developing algorithms to solve problems and, the corresponding process of developing computer programs to express those algorithms. Emphasis will be placed on the development of fundamental computational skills through program design and implementation.

Co-Requisite: CISC 1810 R01 Intro to Computer Science Lab

Class Meeting Time and Location:
Monday/Thursday, 4:00 – 5:15 p.m. Room # 342

Lab Meeting Time and Location:
Wednesday, 2:30 – 3:45 p.m. Room # 331

Instructor: Michael Kadri
Office: RH JMH #340
email: kadri [at] cis [dot] fordham [dot] edu
Office Hours: Monday, Thursday 3:00 – 4:00 p.m.

Objectives:
To introduce students to the field of Computer Science and the software development process whilst building a foundation in the Python programming language.

Outcomes:
A student who successfully completes this course will:
✓ understand and apply the concept of algorithmic problem solving
✓ understand the basic principles of software development.
✓ be able to develop a program using Python principles and constructs.
✓ become more aware of the uses of software within Computer Science.

Textbooks:
To be discussed in class.

Grading:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework &amp; Class Participation</td>
<td>10%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>Programming Labs</td>
<td>20%</td>
</tr>
<tr>
<td>{Documentation (20%), Algorithm &amp; Program Correctness (40%), Style (20%), Quality of output (20%)}</td>
<td></td>
</tr>
<tr>
<td>Mid-term Lab &amp; Exam:</td>
<td>30%</td>
</tr>
<tr>
<td>Final Lab &amp; Exam:</td>
<td>30%</td>
</tr>
</tbody>
</table>

These percentages are guidelines for both the student and instructor and may be
modified by the instructor to better reflect the circumstances of the course.

Resources:
Computer labs: CIS Dept. Windows and Linux Labs LL 812 (JMH 330/1 at Rose Hill)

Blackboard: We will use Blackboard for course material and assignments; you may also use the course Discussion Board as a question and answer forum for course topics and assignments. Please note that you may not post solutions to assignments or an entire program; however, you are encouraged to discuss assignments and any related questions.

Additional Resources:
How to Think Like a Computer Scientist: Interactive Edition
Programming for Everybody (Getting Started with Python | Coursera
How to Think Like a Computer Scientist: Learning with Python 3
This text is freely available online; it can also be downloaded as a PDF.

Attendance:
Class attendance is mandatory. Attendance will be taken at the beginning of each class. If there are class, assignment, or exam dates that conflict with athletic schedules or religious observance please inform the instructor during the first full week of class.

Homework:
In general, there will be homework assigned at the beginning of each new topic covered. It will be in your best interest to complete all homework assignments in a timely manner. Although the total time you spend on homework will vary, you should plan to set aside about six hours each week on homework and assignments for this course.

Laboratory Assignments:
In addition to our regular class time we will be meeting once a week in the Computer Science laboratory to work on programming assignments. Failure to attend labs and complete laboratory assignments will cause at least a letter grade reduction to the individual’s final grade. Late assignments will be accepted with a penalty of 10% of the grade for each late day.

Exams:
Students will be given a laboratory assignment as well as a written examination for both the midterm and the final exam. The laboratory assignment will require the students to complete a Python program during a scheduled lab time. The written exam will take place during class time and test the students on academic material covered from the beginning of the course.

Exam Make-up Policy: There will be no make-up exams given after the exam date. If you know in advance that you will have to miss an exam, you must check with me (in advance) to avoid getting a zero for that exam. In case of illness on an exam date, please contact me as
soon as possible, so that appropriate arrangements can be made.

**Academic Integrity:**
All homework and programming assignments handed in must be your own. Sharing or copying of program code will result in a grade of "zero" for all those involved. Although the sharing of ideas and general computer skills outside of the classroom is encouraged, giving your work and/or submitting someone else’s code as yours will not be tolerated. Official university policy will be followed in such cases of academic dishonesty.

**General Info:**
A student failing all of the exams and or not successfully completing the projects cannot pass this course. Additionally, failing to complete homework and/or computer projects by deadlines set by the instructor can and will cause a reduction to the individual’s final grade.

**Disability:**
Under the Americans with Disabilities Act and Section 504 of the Vocational Rehabilitation Act of 1973, all students, with or without disabilities, are entitled to equal access to the programs and activities of Fordham University. If you believe that you have a disability that may interfere with your ability to participate in the activities, coursework, or assessment of the object of this course, you may be entitled to accommodations. Please schedule a meeting to speak with someone at the Office of Disability Services (Rose Hill - O’Hare Hall, Lower Level, x0655 or at Lincoln Center – Room 207, x6282).

**Course Topics**
The list of the topics covered in the class that follows may not be complete. We may cover more or less, depending on the pace at which we will be able to proceed. The topics to be covered include:

- Introduction to Computers and Programming
  - Hardware and Software
  - How Computers Store Data
  - How a Program Works

- Input, Processing, and Output
  - Displaying Output with the print Function
  - Comments
  - Variables
  - Reading Input from the Keyboard
  - Performing Calculations

- Simple Functions
  - Defining and Calling a Function
  - Designing a Program to Use Functions
  - Local Variables
• Passing Arguments to Functions
• Global Variables and Global Constants

Decision Structures and Boolean Logic
• The if Statement
• The if-else Statement
• Nested Decision Structures and the if-elif-else Statement
• Logical Operators
• Boolean Variables

Repetition Structures
• The while Loop: a Condition-Controlled Loop
• The for Loop: a Count-Controlled Loop
• Calculating a Running Total
• Sentinels
• Input Validation Loops
• Nested Loops

Value-Returning Functions and Modules
• Generating Random Numbers
• Writing Your Own Value-Returning Functions
• The math Module
• Storing Functions in Modules

Files and Exceptions
• Introduction to File Input and Output
• Using Loops to Process Files
• Processing Records
• Exceptions

Lists and Tuples
• Sequences
• List Slicing
• Finding Items in Lists with the in Operator
• List Methods and Useful Built-in Functions
• Copying Lists
• Processing Lists
• Two-Dimensional Lists
• Tuples

Strings
• Basic String Operations
• String Slicing
• Testing, Searching, and Manipulating Strings

Dictionaries and Sets
• Dictionaries
• Sets

Classes and Object-Oriented Programming
• Procedural and Object-Oriented Programming
• Classes
• Working with Instances
• Techniques for Designing Classes

Inheritance
• Introduction to Inheritance
• Polymorphism

Recursion
• Introduction to Recursion
• Problem Solving with Recursion