

**BRONX COMMUNITY COLLEGE**  
**of The City University of New York**

**DEPARTMENT OF MATHEMATICS and COMPUTER SCIENCE**

**CSI 31 Final Exam Study Guide**

PART 1

(Answer the True/False and Multiple Choice questions)

You will be given 5-8 questions

PART 2

Be ready to state what does the following piece of a program do when embedded in a correct program.

(a)

```
a = [ 1, 2, 3, 4, 5, 6]
b = [ -1, 2, -3, 4, -5, 6]
c = [0]
for i in range(6):
    c.append(a[i]*b[i])
s = 0
for i in range(7):
    s = s + c[i]

print('c={0:s}, s={1:d}'.format(c,s))
```

In addition, to answering the questions above, answer the following question:

Explain what happens in the loops, and what are the values of  $c$  and  $s$ , and what is displayed in the Python shell.

(b) see in-class assignments (Lectures 3, 4, 6, and 7)

PART 3

Be ready to find and fix syntax errors in the given piece of program.

For example, find and fix **four** syntax errors:

```
a,b = 10;2
for i in range(b)
    b = 2a + b - 3
    if b > 15
        b = b-5
print("b, a") # display values of b and a
```

**See also:**

- Midterm Exam Outline (Part 3, problem 2)
- in-class assignments (Lectures 5, 12, 15, 16, 17, and 20)

PART 4

(a) Given a code fragment, draw the *flowchart*  
See in-class assignment from Lectures 7, 16, and Midterm Exam.

(b) Given a *flowchart*, be ready to write implementation  
See in-class assignment from Lecture 7, and Midterm Exam.

(c) Be ready to draw the *class diagram* given a class definition  
see in-class assignment from Lecture 20.

(d) Be ready to define a class given a *class diagram*  
see in-class assignment from Lectures 20 and 21.

(e) Be ready to draw a *structure chart* (*module hierarchy chart*) given a program.

PART 5

review work with strings, lists, and dictionaries

review classes

review functions and test functions (unit tests)

review conditionals (see in-class work of Lecture 17)

review top-bottom design and bottom-up implementation

review prototyping and spiral development

review work with files

take a look at the Appendix A (Python Quick Reference) - very useful!