

BRONX COMMUNITY COLLEGE
of The City University of New York

DEPARTMENT OF MATHEMATICS and COMPUTER SCIENCE

CSI 32

Midterm Exam Sample Questions

The Midterm exam will consist of two parts:

Part 1: without use of computer (cheat sheets can be used)

Part 2: with use of computer

Here are examples of questions for Part 1:

1. *class diagrams/class descriptions*: see problem 1.15 (has solution at the end of the book) and problem 1.19 (HW#3)
2. *sequence diagrams*: see problem 1.20 (HW#3) and problem 1.22 (HW#3, although you will be given class diagrams or class descriptions to work with)
3. *inheritance/class diagrams/class hierarchy*: see problem 1.25 (has solution at the end of the book) and problem 1.27 (HW#3)
4. *inheritance/class hierarchy*: See problem 1.28 (has solution at the end of the book) and problem 1.31 (HW#3)
5. Draw an evaluation tree for the following statements:
 - 1) `s.count('a') <= 12 or (r1 - r2 / 4 + 6) > 0`
 - 2) see exercise 2.29 on page 85 (HW#5)
 - 3) $23 + 10(2^2 * 3 - \sqrt{64})$ (HW#5)
6. Explain what does the given piece of code do (in general) - what are the values of v_1 and v_2 after its execution. Then explain what happens at each iteration, give values of variables at each iteration and explain how they change.

Draw a flowchart.

```
v1 = 15
v2 = 54
while v1 < v2:
    v1 -= 1
    v2 = v2/2
print(v1, v2)
```

7. Predict what will be the elements (in *fractional form*) of the list L (without using Python Interpreter)

```
L = [ 1/(i*i+1) for i in range(1,14,2) if i < 12]
```

8. For the following piece of code:

```
x = input("Enter a positive decimal number that is less than 1000:")
try:
    x = eval(x)
except NameError:
    print("This is not a decimal number.")
    x = False
except TypeError: print("You entered...")

if x == False:
    print("Terminating the program.")
elif 0 < x < 1000:
    print(x,"^2 = ",x*x)
else:
    raise ValueError("You entered a decimal that is not positive or is not less t
```

predict the output if

- a) lk is entered
- b) 6 is entered
- c) -10 is entered

Also check out Exercise 6.4 on page 232 (HW#7).

9. review the *terminology* from **Chapter 6** (see **Glossary 6.6.2**).

Questions like:

In the *Class* definition, what is the name of the *constructor* function/method? can come up.

10. Be ready to fix the syntactic errors in a given program. See Practice 1.6 on page 231 (HW#7)

Here are examples of questions for Part 2:

1. An error occurs when evaluating each of the following expressions. Give the official type of error, as reported by the interpreter, and explain in your own words the precise reason for the error.

(a) `'thanks'.insert(1,'g')`

(b) `2/sqrt(3**2-12)` (assume that math module was imported before)

2. recall our **Fraction** class.

Write the definition of two more methods: negation of a fraction (`__neg__`), and negative reciprocal of a fraction (`n_invert`).

3. Be ready to design and develop a simple class.
4. Be ready to use exception handling (catching and raising) (see lecture 6 and HW#6 - this is Chapter 5 of the book)