CSCE 110: Programming I

Sample Questions for Exam #1

September 24, 2011

Below are sample questions to help you prepare for Exam #1. Make sure you can solve all of these problems by hand. For most of the questions, you can check your answers by typing in the programs and seeing what happens on the computer.

1. Study the following code.

Listing 1: q1.py

```python
# statement 1
if x > 0:
    # statement 2
    pass

elif x < 0:
    # statement 3
    pass

else:
    # statement 4
    pass

# statement 5
```

a) Which of the statements above (1, 2, 3, 4, 5) will be executed if $x < 0$?

b) Which of the statements above will be executed if $x == 0$?

c) Which of the statements above will be executed if $x > 0$?

2. Write a program to have the user input (3) numbers: $(f)r\text{o}m$, $(t)o$, $(i)\text{ncr}e\text{m}\text{en}\text{t}$. Count from $f$ to $t$ in increments of $i$, inclusive of $f$ and $t$. For example, if the input is $f == 2$, $t == 26$, and $i == 4$, the program would output: 2, 6, 10, 14, 18, 22, 26.

3. For problem #2, if you used a `while` loop, rewrite your program using a `for` loop. Otherwise, if you used a `for` loop, rewrite your program using a `while` loop. If you didn’t use a `while` or `for` loop for question #2, then rewrite your program as you must use a loop in order to write your program correctly to handle any combination of $f$, $t$, and $i$ entered by the user.
4. What is the output of the following program?

Listing 2: q5.py

```python
a = "blueberry"
print '1:', a[0]
print '2:', a[2]*4
print '3:', a[1:5]
print '4:', a[-1]
print '5:', a[2+3]
print '6:', a[2:]
```

5. Consider the following program.

Listing 3: q5.py

```python
times = ["9:02","8:45","12:44","14:52"],["9:15","11:44"] # Line 1
for day in times :
    # Line 2
    for time in day :
        # Line 3
        print time
        # Line 4
    print # Line 5
```

   a) List all of the keywords in the program.
   b) Write the order in which the lines in the program are executed? For each line in your order, write exactly what is occurring in the Python program at that time. Be specific.
   c) What is the output?

6. For each variable in the code below, provide its data type.

Listing 4: q6.py

```python
a = {}
b = ""
c = ()
d = ""
e = 1
f = []
g = 1.0
h = True
```

If necessary, you can use the built-in function `type()` to help you determine the datatypes of the variables.
7. Consider the following program.

```python
int_value = int(raw_input("Please enter an integer: ")) # Line 1
result = 0 # Line 2
while int_value > 0:
    if int_value / 3 > 1:
        result = result + 3 # Line 5
    elif int_value / 3 > 0:
        result = result + 4 # Line 7
    else:
        result = result + 1 # Line 8
    int_value = int_value - 2 # Line 9
print "result is", result # Line 11
print "int_value is", int_value # Line 12
```

a) Write the execution order of the lines in this program. Assume that the user types the value 5 at Line #1.

b) Given a user input of 5, what output is produced by Line #11.

c) Given a user input of 5, what output is produced by Line #12.

d) Given a user input of 5, how many times is Line #3 evaluated.

8. Write the following Python programs.

a) Write a program that asks a user to enter an integer. Now, tell them the number of 4s in their integer. For example, if the user types in the integer 345924, your program will report that the number four appears two times. Make sure when you write your program, your output is easy to read.

b) Take the program you wrote in 8(a) and extend it to report back the number of times each of the digits between 0 and 9 appeared in the integer that the user entered. For example, if the user entered 18998311, then you would report that the user’s number contained three 1s, one 3, two 8s, and two 9s. Make sure your output is easy to read.

c) Ask the user tell you how many numbers they would like to enter. Say this number is 5. Then have them input 5 numbers. Next, find the maximum number that was entered by the user. Do not use the `max` or `sort` functions in Python to write this code! Your code will consist of loops and decision-making constructs.

Suppose the user wishes to enter 5 numbers. Let’s say the five numbers entered are 10, 100, 34, 579, and 1. Then, your program will output that the maximum number that is entered by the user is 579. Your program should work regardless of whether the user enters 5 numbers, 10 numbers, 1000 numbers, etc.