Below are sample questions to help you prepare for Exam #2. 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. When writing your programs by hand, please write clearly and make sure that the indenting of your program is clear.

1. The following code is supposed to compute the product of the numbers in a list. For example, `product([3, 5, 2])` should return 30. However, the program contains a logical error. What is the logical error? What line(s) cause the problem? Fix the error.

```
Listing 1: error.py

def product(nums):
    prod = 1
    for n in nums:
        prod = prod * n
    return prod

print product([3, 5, 2])
```

2. Consider the following program.

```
Listing 2: functions.py

n = 10

def foo(x):
    n = 3 + x
    x = n * 2
    return n, x

def bar(m):
    n = 4 + m
    a, b = foo(n)
    print 'The result in bar is %d.'%(a + b)

bar(10)
print 'Goodbye!
```

a) List all of the keywords in the program.

b) Write the order in which the lines in the program are executed?
c) What is the output?

d) For each function, list each of its local variables and the line number(s) in which the local variables appear.

e) List all of the global variables that appear in the program and the line numbers in which they appear.

3. Consider the following Python program.

Listing 3: sets.py

```python
def f1(s1, s2, op):
    if op == 'i':
        result = s1.intersection(s2)
    elif op == 'u':
        result = s1.union(s2)
    elif op == 'd':
        result = s1.difference(s2)
    else:
        result = s1.issubset(s2)
    return result

def main():
    set1 = set(['apple', 5, 'banana', 7])
    set2 = set(['banana', 5])
    print len(set2)  # Line A
    print f1(set1, set2, 'i')  # Line B
    print f1(set2, set1, 'i')  # Line C
    print f1(set1, set2, 'u')  # Line D
    print f1(set2, set1, 'u')  # Line E
    print f1(set1, set2, 'd')  # Line F
    print f1(set2, set1, 'd')  # Line G
    print f1(set1, set2, 'p')  # Line H
    print f1(set2, set1, 'p')  # Line I
```

main()

a) What output does Line A produce?

b) What output does Line B produce?

c) What output does Line C produce?

d) What output does Line D produce?

e) What output does Line E produce?

f) What output does Line F produce?

g) What output does Line G produce?

h) What output does Line H produce?

i) What output does Line I produce?

4. Consider the following functions-and-files-data.txt, which is a text file. It is also a CSV file.
Now consider the following Python program, which is called functions-and-files.py.

```
Listing 5: functions-and-files.py

def f1 (fd , the_list ) :
    count = 0
    for line in fd :
        line_list = line . split (',',)
        result = 0
        for element in line_list :
            result += int (element)
        the_list += [(count , result)]
        count += 1
    return the_list

def f2 (the_list ) :
    val = 0
    for element in the_list :
        if element [1] > val :
            val = element [1]
    return val

def main () :
    fd = open ('functions-and-files-data.txt' )
    result1 = f1 (fd , [] )
    print result1 # Line A
    result2 = f2 (result1 )
    print result2 # Line B
    main ()
```

Assume that functions-and-files-data.txt and functions-and-files.py are in the same directory. Please answer the following questions.

a) What output does Line A produce?

b) What output does Line B produce?

5. Give a brief description of what the program functions-and-files.py does?
6. Consider the following Python program.

Listing 6: myFun.py

```python
def myFun(param1, param2):
    result = ""
    inc = param1
    while param1 < len(param2):
        result = result + param2[param1]
        param1 += inc
    return result

def main():
    raw_inputs = raw_input("Give me an int and a string: ")
    inputs = raw_inputs.strip()
    int_arg, str_arg = inputs.split(' ')
    int_arg = int(int_arg)
    print int_arg
    print str_arg
    print myFun(int_arg, str_arg)

main()
```

a) What is the output by Line C if the user inputs "1 Aggies" (without the quotes) when prompted?

b) What is the output by Line D if the user inputs "1 Aggies" (without the quotes) when prompted?

c) What is the output by Line E if the user inputs "1 Aggies" (without the quotes) when prompted?

d) What is the output by Line E if the user inputs "4 Aggies" (without the quotes) when prompted?

e) What is the output by Line E if the user inputs "0 Aggies" (without the quotes) when prompted?

f) If the user inputs "0 Aggies" (without the quotes) when prompted, what is the effect on the program if the < symbol in Line A is changed to <=? Explain.

g) If the user inputs "0 Aggies" (without the quotes) when prompted, what is the resulting output if Line B is changed to the following?

```
result = result + param2[param1]
```

7. Give a brief description of what the above program does?
8. Consider the following program that uses the `matplotlib` module.

Listing 7: plot.py

```python
import matplotlib.pyplot as plot

def draw_plot(x_axis, y_axis, x_ticks, y_ticks):
    plot.plot(x_axis, y_axis, marker='o', markersize=7.0,
              label='a simple line')
    plot.xlabel('x axis')
    plot.ylabel('y axis')
    plot.title('a simple plot')
    plot.xticks(x_ticks)
    plot.yticks(y_ticks)
    plot.legend()
    plot.grid(True)
    plot.show()

def main():
    step = 1
    # Line A
    x_list = range(5)
    y_list = [3, 2, 5, 0, 2]
    xticks_list = [0, 1, 2, 3, 4]
    yticks_list = range(min(y_list), max(y_list) + 2, step)
    print yticks_list  # Line B
    draw_plot(x_list, y_list, xticks_list, yticks_list)  # Line C

main()
```

   a) What output does Line B produce?

   b) What output does Line C produce? Yep, I want you to be able to draw the plot by hand.

   c) What is the effect on the program if Line A is changed to `step = 2`. Be sure to show the resulting output.

9. Write a function `roll_dice(n)`, which returns the result of rolling n dice. For example, if n is 1, then your function will return the result of 1 die. If n is 10, then your function will return the individual rolls of each of the 10 dice. Thus, your `roll_dice()` function will return a list of values of size n. Next, write a main function which calls your `roll_dice()` function. In the main function, output the values of the odd-numbered dice.

10. Write a Python program that computes the sum of numbers on each line of a CSV formatted-file.

   a) You will read in a CSV file, the user provides the filename. Assume the file exists and is in the same directory (i.e., working directory) as the Python program you are writing. Each line of that file consists of a set of comma separated values. Compute the sum of the values on each line and print them in the format:

      `<line number>`: `<sum>`

   You will also report the highest sum and the line number that contains it.
b) For example, consider the file sample-file.csv, which contains the following.

```
1 1,2,3
2 1,1,1,1
3 4,5,6
4 1,1
```

c) Your program should report:

```
Please enter a filename: sample-file.csv

<line number>: <sum>
1:6
2:4
3:15
4:2

Line 3 has the highest sum of 15.
```

11. As in Question #6, you will analyze a CSV file of numbers. Your goal is to write a program that outputs the duplicate numbers. Consider sample-file.csv shown in Question #6. Your program should report:

```
Please enter a filename: sample-file.csv

The duplicate numbers are:
1
```

12. Write a function called is_vowel(c) to determine whether a character (i.e., a string of length 1) c is a vowel. Your function should return a Boolean value. That is, is_vowel() returns True if c is a vowel and False otherwise.

13. Write a function called is_consonant(c) to determine whether a character c is a consonant. Your function should return a Boolean value. That is, is_consonant() returns True if c is a consonant and False otherwise.

14. Suppose you have the following list of numbers: 1, 3, 5, 29, 4, 17, and 8.

   a) Write a Python program that uses a for loop to increment each value in the list by 5. Print the new list of numbers.
   
   b) Write a Python program to show how to do this operation using the numpy module.