

CAIH 150 Exam Part 1 Overview.

Resources During the Exam

Due to the exam being online, I cannot enforce a closed book, closed computer requirement. Therefore, you may use the tutorial and your computer if you wish. However, the exam is timed, and it is not designed to allow for the extensive use of either the tutorial or your computer. I strongly suggest that you summarize the important notes on two sides of 8.5 x 11 inch paper. You may want to include a summary of operators, string formats, the syntax of a loop statement, how to define and use a dictionary, etc. Write this as you study! I test you on concepts, not memorized rote facts. *Bring* the facts if you like! *Learn* the concepts.

Important note about using Sakai for the exam: When you are writing code, be sure to use the “Rich Text Editor”, or it will not preserve any indentation.

• just change the second line of the code to change the output

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich Text Editor (and character count)

Main topics that may be on the exam

Python Tutorial Chapter 1. See the summary at the end of the chapters.

How the Python topics get used:

1. Part 1: You are given python code segments that generate output (or sometimes, an error). You are asked to change the code to generate different output. You are sometimes asked why the given code generated an error. Problems 1-7, and Problem 17 of this review are indicative of the type of problems that will be on Part 1 of the exam.
2. Part 2: Write a segment of code translating an idea (specifications) into Python, or put a few steps together. Sometimes, you will be given the description of a function, and you are required to write the function. Problems 18 -25 of this review are indicative of the type of problems that will be on Part 2 of the exam.

Read the following before looking at either the problems or the solutions!

1. Study first, gathering your written notes. Look at the chapter summary and start by filling in any holes. Then look at the sample problems. The sample problems cannot give complete coverage, and if you look at them first, you are likely to study just these points, and will not get an idea how well you are prepared in general.
2. Do not look at the answers until you have fully studied and tried the problems and gotten *help* getting over rough spots in the problems if you need it! Looking at the answers before this time makes the problems be just a few more displayed examples, rather than an opportunity to actively learn by doing and check out where you are. The *doing* is likely to help you be able to *do* again on a test.

Sample problems start on the next page.

Review Problems for Chapter 1 using Python 3.8+

For questions 1 – 7, 17 and 18, you have examples of the type of problem where you are given code (and usually the output), and you have to change the code to generate new given output.

For questions 8-16, you have examples of code that will help you for either Part 1 (change the code for new output) or Part 2 (given specifications, write new code). There are no questions similar to Questions 8 – 16 on the exam. That is because they ask you to predict the output of the code that is given. We can't have problems like that on an online exam, because they you could just run them on your computer. However, the skills that are shown in Questions 8-16 in this review are also incorporated into the exam in other formats.

1. Consider the following code:

```
x = 5      #1
y = x + 3  #2
x = x - 1  #3
z = 10     #4
x = x + z  #5
print('x: {}, y: {}, z: {}'.format(x, y, z)) #6
```

If you want to review how the original code works, before you change it for the new output, see the Answers section at the end of the document.

This question is a little difficult...

Currently, the output is:
x: 14, y: 8, z: 10

Change the code so that the output is:
x: 12, y: 10, z: 8

Only two lines of code must be changed

Answer:

- change line #2 to `y = x + 5`
- change line #4 to `z=8`

2. Consider the following Python code:

```
print(14//4, 14%4, 14.0/4)
```

Currently, the output is:
3 2 3.5

Change the code to generate the following output:
3.5 2 3.5

Note that you did not have to change anything for the second and third values, because the output is the same.

Answer:

```
print(14/4, 14%4, 14.0/4)
```

3. Consider the following Python code:

```
print(2*'No' + 3* '!') # prints NoNo!!!
```

Precedence of operations

Change the code to generate the following output:

```
No!!!No!!! # Answer: print(2 * ('No' + 3* '!'))
```

4. Understand what is printed by the Python code. Be careful: Note the backslashes:

```
print('how\nis it\nnow')
```

5. Consider the following Python code?

```
for z in [2, 4, 7, 9]:  
    print(z - 1)
```

Currently, the output generated is:

```
1  
3  
6  
8
```

Change ONLY the print statement to generate the following output:

```
3  
5  
8  
10
```

Answer: `print(z+1)`

6. Consider the following Python code:

```
print('2' + '3')           # '23'  
print(2 + 3)               # 5  
print('2' + 3)            # ERROR
```

You will be asked if something generates an error, and you will have to explain why it generated an error.

7. Consider the following Python code:

```
def f1():  
    print('Hi')  
def f2():  
    print('Lo')
```

```
f2()  
f1()  
f1()
```

Currently, the output generated is:

```
Lo  
Hi  
Hi
```

You may want to review the difference between a function definition and a function call.

Change the code to print the following:

```
Lo  
Hi  
Lo
```

Answer:

```
f2()  
f1()  
f2()
```

There are no “What is printed?” questions on the exam. However, you may want to review problems 8 – 16 on this review as some of the same concepts and skills are included in other types of questions on the exam. The answers to these questions are in the Answers section at the end of this document.

8. What is printed by the Python code?

```
def func():  
    print('Yes')  
  
print('No')  
func()
```

9. What is printed by the Python code?

```
def func(x):  
    print(2*x)  
  
func(5)  
func(4)
```

10. What is printed by the Python code?

```
def func(x):  
    return x - 1  
  
print(func(3) * func(5))
```

11. What is printed by the Python code?

```
n = 3 #1  
for x in [2, 5, 8]: #2  
    n = n + x #3  
    print(n) #4
```

12. What is printed by the Python code?

```
print(list(range(3)))
```

13. What is printed by the Python code?

```
for i in range(3):  
    print('Hello again!')
```

14. What is printed by the Python code?

```
for i in range(4):  
    print(i)
```

15. What is printed by the Python code?

```
def s(x): #1  
    return x*x #2  
  
for n in [1, 2, 10]: #3  
    print(s(n)) #4
```

16. What is printed by the Python code?

```
def s(x): #1  
    return x*x #2  
  
tot = 0 #3  
for n in [1, 3, 5]: #4  
    tot = tot + s(n) #5  
    print(tot) #6
```

17. Consider the following Python code?

```
x = 2.5679
y = 9.0
print('Answers {:.3f} and {:.3f}'.format(x, y))
```

Currently, the output generate is:
Answers 2.568 and 9.000

Change the print statement to print the following:
Answers 2.57 and 9.0000

```
# The new print statement is:
# print('Answers {:.2f} and {:.4f}'.format(x, y))
```

18. Consider the following Python code?

```
d = dict()
d['left'] = '<<'
d['right'] = '>>'
print('{left} and {right} or {right} and {left}'.format(**d))
```

Currently, the output generated is:
<< and >> or >> and <<

Change the print statement to print:
>> and >> or << and <<

The new print statement is:
print('{right} and {right} or
{left} and
{left}'.format(**d))

- You may want to review `madlib.py`—we also create a dictionary there and add some entries to the dictionary.
- Unlike `madlib.py`, you do not have to create any functions here.
- Recall that `format(**d)` means “use the dictionary named `d` to format the string”, meaning, “wherever you see a key in the braces of the string, substitute the value for that key from the dictionary `d`”.

Answers to Questions 19 – 25 are in the Answers section at the end of this document..

19. Write a Python program that prompts the user for two numbers, reads them in, and prints out the product, labeled.

20. Given a string `s`, write a short expression for a string that includes `s` repeated five times.

21. Suppose you know `x` is an integer and `ys` is a string representing an integer. For instance, `x` is 3 and `ys` is '24'. Write code to print out the arithmetic sum of the two. In the example case, 27 would be printed.

22. Suppose you are given a list of words, `wordList`. Write Python code that will write one line for each word, repeating that word twice. For example if `wordList` is ['Jose', 'Sue', 'Ivan'], then your code would print

```
Jose Jose
Sue Sue
Ivan Ivan
```

23. Write code to create a Python dictionary (the `dict` type). Add two entries to the dictionary: Associate the key 'name' with the value 'Juan', and associate the key 'phone' with '508-1234'

24. Complete the code for the following function so it matches its documentation:

```
def doubleList(numberList):  
    ''' For each of the numbers in the list numberList, print a line  
        containing twice the original number. For example,  
        doubleList([3, 1, 5]) would print  
            6  
            2  
            10  
        '''
```

Q24: The multi-line string is documentation. The example given is just one example of many values that can be used to call the doubleList function. Do NOT call the function in the function def.

25. Assuming a function process is already defined, write two lines of code, using a **for**-loop, that is equivalent to the following:

```
process('Joe')  
process('Sue')  
process('Ann')  
process('Yan')
```

Q25: Anytime that you have a sequence of items that require the identical processing, you may consider putting the items in a list, and using that list in a for...loop

Answers start on the next page

Below are answers to the unmodified Q1 – Q7, before you modify for new output.

Review Problem Answers

1. x: 14, y: 8, z: 10

Here like in all the answers, the details are not required in an exam, but they might help you get partial credit if you make a mistake somewhere in the middle! Details:

line	x	y	z	comment
1	5	-	-	
2		8		8=5+3
3	4			4=5-1
4			10	
5	14			14=10+4
6				substitutes into format and prints result above

2. 3 2 3.5

14 divided by 4 is 14//4=3 with a remainder of 14%4=2. Because of the single '/' in last part, the result has a decimal point.

3. NoNo!!!
No!!!No!!!

4. how
is it
now

In a string literal \n means newline.

5. 1
3
6
8

Print one less than each number in the list.

6. 23

7. Lo
Hi
Hi

First the functions are remembered. Afterward they are called in the order given.

8. No
Yes

Below are answers to the unmodified Q8 – Q16.

First the function is remembered. It is only called after 'No' is printed.

9. 10
8

First the function is remembered. Afterward it is called with x = 5, returning 10=2*5. Finally it is called with x=4, returning 8 = 2*4.

10. 8

(3-1)*(5-1) = 2*4 = 8. The function is called twice and the results are combined.

11. 18

details:

short version: 3+2+5+8 = 18

long version:

line	n	x	comment
1	3	-	
2		2	first value in list
3	5		5=3+2
2		5	second value in list
3	10		10=5+5
2		8	last value in list
3	18		18=10+8
2			done with list and loop
4			prints 18

12. [0, 1, 2]

start with 0, ends before 3

13. Hello again!
Hello again!
Hello again!

The sequence range(3) has 3 elements so the loop is repeated 3 times. (Simple repeat loop: variable ignored.)

14. 0
1
2
3

range(4) contains 0, 1, 2, 3

15. 1
4
100

Evaluates s, the squaring function, for each element in the list, and prints the results.

16. 35 # 0 + 1*1 + 3*3 + 5*5

details:

```
line tot n x comment
1-2      definition
3        0
4        1 first value in list
5        evaluate s(1), returns 1 = 1*1
         1 so tot = tot+1 = 0+1 = 1
4        3 next value in list
5        evaluate s(3), returns 9 = 3*3 10
         so tot = tot+9 = 1+9 = 10
4        5 last value in list
5        evaluate s(5), returns 25 = 5*5 35
         so tot = tot+25 = 10+25 = 35
4        done with list and loop
6        prints 35
```

17. Answers 2.568 and 9.000

Substitutions into format string with floating point formats. Both show 3 decimal places because of the 3's in the floating point formats. Results are also *rounded* automatically: 2.568, not 2.567.

18. << and >> or >> and <<

These are the answers to review questions Q18 – Q25, where you have to write new code. These are similar to what you have to do for Part 2 of the exam.

Formats with {key} substitute strings from the

dictionary 19.

```
x = int(input('Enter a number: ')) # or some such prompt
y = int(input("Enter another number: ")) # or some such prompt
print('The product is ', x*y) # or some such label
```

20. s*5 # or: s+s+s+s+s

Q20: You were NOT asked to write a program or a function, nor to assign a value to S. You are “given” s. Don’t overthink or overdo the problems.

21. print(x + int(ys))

Q21: You may choose to break this down into steps:

```
22. for word in wordlist: # variable word is arbitrary
    print(word, word) # but must match here!
```

```
yInt = int(ys)
sum = x + yInt
print(sum)
```

23. d = dict() # name d is arbitrary, but match it in the
nextlines

```
d['name'] = 'Juan'
d['phone'] = '508-1234'
```

Q22: You can use any variable name you want instead of “word”
But you MUST use wordlist, because the problem states that as given.

```
24. def doubleList(numberList):
    ''' skip repeating docs...'''
    for n in numberList:
        print(2*n)
```

Q24: Notice that you do NOT call doubleList inside of the doubleList function. You do use the numberList parameter in your for...loop

```
25. for name in ['Joe', 'Sue', 'Ann', 'Yan']:
    process(name)
```

Q25: Anytime that you have a sequence of items that require the identical processing, you may consider putting the items in a list, and using that list in a for...loop