# CSSE1001: Sem. 1 2012 exam answers

## [UQAttic](http://uqattic.net/)

## Get more out of your study time. Join UQAttic's Semester 2, 2012 revision chat.

#### [**Other exam papers**](https://docs.google.com/folder/d/0B_gnBD5zp3rqT2hrLWhvd2Rkc1U/edit)

### Read this page first!

### Please contribute to these documents.

### If you're looking for an effective way to familiarise yourself with the course material, you can't go past collaborating with fellow students. We have laboured to put these up, and so at the very least point out where you think we are wrong!

### You'll get more out of the course, you'll do better in the exam, and other students will benefit from your input as well.

### To get editing permissions, simply go to the [chatroom](http://mibbit.com/?channel=%23attic&server=irc.oftc.net) and provide us with your Google Account address.

### Style.

### If you're unsure of your answer, highlight your answer text then hit Ctrl+Alt+M to create a comment beside the text. Once you're satisfied with the answer, click the "Resolve" button on the comment.

### If you want some extra explanation from someone else on their answer, highlight the other person's answer and repeat the procedure above.

### If you are confident that your answer is correct, then just type it in red below each question.

### Communicate.

### Head over to [uqattic.net](http://uqattic.net/) and click "Chat Now!". You'll find a chatroom full of students just like you. Talk about a revision document (like this one) or swap prep tips.

### If you have your own IRC client, point it to irc.uqattic.net, port 6667, channel #attic.

### You can disable automatic email alerts of comment discussions via the "Discussions" button.

You should look at the exam questions before reading these answers: <http://www.library.uq.edu.au/exams/papers.php?stub=csse1001>

1. What does the expression 3.0 + 7/2 evaluate to?

(a) 6.5

(b) 6.0

(c) 5.0

(d) 5

Answer: b (since 7/2 is 3, then 3.0+3 is 6)

2. What does the expression 3.0 - 7/2 evaluate to?

(a) -2.0

(b) -0.5

(c) 0.0

(d) 0

Answer: c

3. What does the expression 2 \* ['1', '3'] evaluate to?

(a) [2, 6]

(b) ['2', '6']

(c) ['1', '1', '3', '3']

(d) ['1', '3', '1', '3']

(e) Error

Answer: d

4. What does the expression '2' + ['1', '3'] evaluate to?

(a) [3, 5]

(b) ['3', '5']

(c) ['21', '23']

(d) ['2', '1', '3']

(e) Error

Answer: e (can’t add a string and a list)

5. What does the expression list('2') + ['1', '3'] evaluate to?

(a) [3, 5]

(b) ['3', '5']

(c) ['21', '23']

(d) ['2', '1', '3']

(e) Error

Answer: d (calling list(‘2’) makes a list of all the characters in the string ‘2’, i.e. [‘2’].)

6. After the assignment x = 'Very Naughty Boy', which of the following assigns 'u' to the variable y?

(a) y = x[7]

(b) y = x[6]

(c) y = x[-8]

(d) None of the above.

Answer: a (counting from 0, the u is at position 7)

7. After the assignment x = 'Very Naughty Boy', which of the following assigns 'aug' to the variable y?

(a) y = x[5 : 8]

(b) y = x[5 : 9]

(c) y = x[6 : 9]

(d) y = x[6 : 10]

(e) None of the above

Answer: c (the slice [6:9] goes from position 6 (‘a’) up to but not including position 9 (‘h’))

8. After the assignment x = 'Very Naughty Boy', which of the following assigns 'Boy' to the variable y?

(a) y = x[-2 : -1]

(b) y = x[-3 : -1]

(c) y = x[-2: ]

(d) y = x[-3 : ]

(e) None of the above

Answer: d (the slice [-3:] goes from the third-last character to the end)

9. After the assignment x = 'Very Naughty Boy', which of the following assigns 'yth' to the variable y?

(a) y = x[-4 : -8: -1]

(b) y = x[-5: -8 : -1]

(c) y = x[10 : 7 : -1]

(d) y = x[11 : 7 : -1]

(e) None of the above

Answer: b (the slice goes from position -5 (‘y’) down to but not including -8 (‘g’), in steps of -1)

10. After the assignment x = '1 2 3 4 5', what does the expression

''.join(x.split(' ', 2)) evaluate to?

(a) '12'

(b) '1 2'

(c) '12 3 4 5'

(d) '123 4 5'

(e) Error

Answer: d (the split gives back [‘1’, ‘2’, ‘3 4 5’], then joining it with ‘’ gives ‘123 4 5’)

11. What is the value of a after the following is evaluated?

def f(x, y):

y.append(x)

return y

a = [1,2]

a = f(3,a) + a

(a) [1, 2, 1, 2]

(b) [1, 2, 1, 2, 3]

(c) [1, 2, 3, 1, 2]

(d) [1, 2, 3, 1, 2, 3]

(e) Error

Answer: d (the function calls a.append(3), so a == [1,2,3], then returns a, and adds on a.)

12. Which of the following statements about dictionaries is true?

(a) Dictionary keys can be any type.

(b) Only numbers or strings can be dictionary keys.

(c) Dictionary keys must be immutable types.

(d) Both dictionary keys and values must be immutable types.

(e) If the dictionary keys are mutable then the corresponding values must be immutable.

Answer: c (you can’t have lists or dictionaries as keys of a dictionary, for example. There’s no restriction on what the values can be.)

13. What is the value of y after the following is evaluated?

d = {1: 'a', 3:'b', 5:'c', 7:'d'}

y = [d.get(3), d.get(0)]

(a) ['b', 'a']

(b) ['b', '']

(c) ['b', None]

(d) Error

Answer: c (d.get(0) returns None when 0 is not in d)

14. What is the value of y after the following is evaluated?

d = {1: 'a', 3:'b', 5:'c', 7:'d'}

y = [d.get(3, ''), d.get(0, '')]

(a) ['b', 'a']

(b) ['b', '']

(c) ['b', None]

(d) Error

Answer: b (d.get(0, ‘’) returns the second argument ‘’ when 0 is not in d)

The next 3 questions refer to the following deﬁnition. Recall that y.pop() returns the last element of the list and removes the last element from the list.

def f(x, y):

return x + [y.pop(), x.pop()]

15. What is the value of w after the following is evaluated?

a = [4,5]

b = (1,2,3)

w = f(a,b)

(a) (4, 5, 3, 5)

(b) [4, 5, 3, 5]

(c) (4, 5, 1, 4)

(d) [4, 5, 1, 4]

(e) Error

Answer: e (you can’t call .pop() on a tuple)

16. What is the value of w after the following is evaluated?

a = [4,5]

b = [1,2,3]

w = f(a,b)

(a) [4, 5, 3, 5]

(b) [4, 5, 1, 4]

(c) [4, 3, 5]

(d) [4, 1, 5]

(e) Error

Answer: c (pop the 3 from b and the 5 from a, then return a + [3, 5] == [4, 3, 5])

17. What is the value of w after the following is evaluated?

a = [1,2,3]

w = f(a, a)

(a) [1, 2, 3, 1, 2]

(b) [1, 2, 3, 2, 1]

(c) [1, 2, 3]

(d) [1, 3, 2]

(e) Error

Answer: d (pop the 3 from a, then the 2 from a, then return a + [3, 2] == [1, 3, 2])

The next 3 questions refer to the following deﬁnition.

def s(xs,n,m):

t = []

while n < m:

t.append(xs[n])

n += 2

return t

18. What is the value of x after the following is evaluated?

x = s([1,2,3,4,5], 0, 4)

(a) []

(b) [1, 3]

(c) [1, 3, 5]

(d) [1, 2, 3, 4]

(e) Error

Answer: b (first time through the loop, n==0, so we append 1; second time through, n==2, so we append 3; then n==4, which is not less than 4, so we finish.)

19. What is the value of x after the following is evaluated?

x = s([1,2,3,4,5], 4, 0)

(a) []

(b) [1, 3]

(c) [1, 3, 5]

(d) [1, 2, 3, 4]

(e) Error

Answer: a (as we reach the loop the first time, n is not less than m, so the loop never happens)

20. What is the value of x after the following is evaluated?

x = s([1,2,3,4,5], 2, 4)

(a) []

(b) [2]

(c) [2, 4]

(d) [2, 3, 4]

(e) Error

Answer:

It should be [3], which is not an option. (This was an error in the exam question).

First time through the loop, n == 2, so we append 3 (i.e. xs[2]). Then, n == 4, which is not less than 4, so we finish.

The next 3 questions refer to the following deﬁnition. Recall that enumerate(xs) (where xs is iterable) is an iterator that produces pairs of indices and elements of xs.

def z(xs, ys):

t = []

for i, x in enumerate(xs):

t.append((x, ys[:i+1]))

return t

**Note**: there was an error in the exam, the line inside the loop should be t.append((x, ys[i+1:]))

21. What is the value of zs after the following is evaluated?

zs = z([1,2,3], [4,5])

(a) [1, 4, 2, 5, 3]

(b) [(1, 4), (2, 5)]

(c) [(1, [5]), (2, [])]

(d) [(1, [5]), (2, []), (3, [])]

(e) Error

Answer: d (a list of pairs, the first element comes from [1,2,3]. d is the only one that fits this)

22. What is the value of zs after the following is evaluated?

zs = z([1,2,3], [4,5,6])

(a) [1, 4, 2, 5, 3, 6]

(b) [(1, 4), (2, 4), (3, 6)]

(c) [(1, [5, 6]), (2, [6]), (3, [])]

(d) [(1, [5, 6]), (2, [6])]

(e) Error

Answer: c (a list of pairs, the first element comes from [1,2,3], and the second one is a list. option c is the only one that fits this)

23. What is the value of zs after the following is evaluated?

zs = z([1,2], [4,5,6])

(a) [1, 4, 2, 5]

(b) [(1, 4), (2, 5)]

(c) [(1, [5]), (2, [])]

(d) [(1, [5, 6]), (2, [6])]

(e) Error

Answer: d (a list of pairs, the first element comes from [1,2], and the second one is a list. option d is the only one that fits this)

The next 3 questions refer to the following deﬁnition that is missing three lines of code. This function checks for the correctness of the indentation of a supplied Python ﬁle. For simplicity we assume the ﬁle being checked has no triple quoted or # comments. For the indentation to be correct, each line should start with a multiple of 4 space characters (not tabs) and be consistent with the required indentation of a Python program. Note: A semi-colon can be used to join two lines of code into one line.

def indent(string):

"""Return the number of space characters at the beginning

of string.

indent(str) -> int

"""

return len(string) - len(string.lstrip())

def check\_indents(filename):

"""Check the indentation of the given Python file.

Returns the line number of the first indentation error

or -1 if no indentation errors are present.

check\_indents(str) -> int

"""

handle = open(filename, 'rU')

lines = handle.readlines()

handle.close()

curr\_ind = 0

just\_ind = False

for i in range(len(lines)):

line = lines[i].rstrip()

if line == '':

continue

line\_ind = indent(line)

if line\_ind % 4 != 0: return i

line\_ind /= 4

## line 1 ##

return i

if line.endswith(':'):

## line 2 ##

else:

## line 3 ##

return -1

24. What is the required code for ## line 1 ##?

(a) if just\_ind and line\_ind != curr\_ind:

(b) if just\_ind or line\_ind != curr\_ind:

(c) if line\_ind > curr\_ind or just\_ind and line\_ind != curr\_ind:

(d) if line\_ind > curr\_ind or just\_ind or line\_ind != curr\_ind:

(e) if line\_ind > curr\_ind

Answer: c (the line is incorrect if there is more indenting than we were expecting, or we have just indented and the indentation is not what we were expecting)

25. What is the required code for ## line 2 ##?

(a) curr\_ind = line\_ind + 1; just\_ind = True

(b) curr\_ind = line\_ind - 1; just\_ind = False

(c) curr\_ind = line\_ind + 1

(d) just\_ind = True

(e) just\_ind = False

Answer: a (what we should be expecting for the next line is one level more indentation than this line, and set the just\_ind flag so that the if statement condition from Q24 picks this up)

26. What is the required code for ## line 3 ##?

(a) curr\_ind = line\_ind - 1; just\_ind = True

(b) curr\_ind = line\_ind + 1; just\_ind = False

(c) curr\_ind = line\_ind - 1

(d) just\_ind = True

(e) just\_ind = False

Answer: e (the next line shouldn’t be indented, so we remove the just\_ind flag)

^ I don’t think this correct. I think none of the options are correct, and that it should be

curr\_ind = line\_ind ; just\_ind = True

The next three questions refer to the following partial deﬁnition of a PhoneBook class for storing phone numbers of people.

class PhoneBook:

def \_\_init\_\_(self):

"""A dictionary whose keys are names and whose values are phone

numbers is used to represent the phone book. The dictionary is

initially empty"""

self.\_numbers = {}

def add\_number(self, name, number):

"""Update the phone book by adding the given name and phone number.

add\_number(str, str) -> None""

## line 1 ##

def get\_number(self, name):

"""Return the phone number of the person with the given

name. Return None if that person is not in the phone book.

get\_number(str) -> str"""

## line 2 ##

Assume that the following has been evaluated.

my\_numbers = PhoneBook()

27. What is the required code for ## line 1 ##?

(a) numbers[name] = number

(b) \_numbers[name] = number

(c) self.numbers[name] = number

(d) self.\_numbers[name] = number

(e) More than one of the above is correct.

Answer: d (the dictionary we want to update is called self.\_numbers)

28. What is the required code for ## line 2 ##?

(a) print self.\_numbers[name]

(b) return self.\_numbers[name]

(c) print self.\_numbers.get(name)

(d) return self.\_numbers.get(name)

(e) More than one of the above is correct.

Answer: d (the docstring says the function should return (not print). If the person isn’t in the phone book, it should return None - .get() does this)

29. Which of the following correctly updates the phone book by adding Fred whose number is

5551234?

(a) add\_number(my\_numbers, 'Fred', '5551234')

(b) my\_numbers.add\_number('Fred', '5551234')

(c) my\_numbers.add\_number['Fred'] = '5551234'

(d) my\_numbers.\_add\_number['Fred'] = '5551234'

(e) More than one of the above is correct.

Answer: b method call syntax is: object\_name.method(arguments)

The ﬁve questions on the following page refer to the class deﬁnitions and assignments given below.

class A:

def \_\_init\_\_(self, x):

self.x = x

def f(self, x):

return self.g(x)-1

def g(self, x):

return x+1

class B(A):

def g(self, y):

return y + self.x

class C(B):

def \_\_init\_\_(self, x, y):

B.\_\_init\_\_(self, x)

self.y = y

def f(self, x):

return A.g(self,x) + self.y

class D(C):

def \_\_init\_\_(self, x, y):

C.\_\_init\_\_(self, x, y)

self.y = y + 2

def g(self, x):

return x + self.x

a = A(2)

b = B(2)

c = C(2, 2)

d = D(2, 0)

30. What does the expression b.g(3) evaluate to?

(a) 3

(b) 4

(c) 5

(d) 6

(e) Error

Answer: c -- b.g(3) calls B.g, which returns 3 (y) + 2 (self.x)

31. What does the expression a.f(3) evaluate to?

(a) 3

(b) 4

(c) 5

(d) 6

(e) Error

Answer: a -- a.f(3) returns a.g(3) - 1. a.g(3) returns 3+1==4, so a.f(3) returns 4-1==3

32. What does the expression b.f(3) evaluate to?

(a) 3

(b) 4

(c) 5

(d) 6

(e) Error

Answer: b -- b.f(3) calls the f of A, which returns b.g(3) - 1. From Q30, this is 5, and 5-1==4.

33. What does the expression c.f(3) evaluate to?

(a) 3

(b) 4

(c) 5

(d) 6

(e) Error

Answer: d -- c.f(3) returns A.g(c,3) + c.y. Calling A’s g method on 3 returns 4; c.y is 2; 4+2=6.

34. What does the expression d.f(3) evaluate to?

(a) 3

(b) 4

(c) 5

(d) 6

(e) Error

Answer: d -- d.f(3) does the same as c.f(3). Note that D.\_\_init\_\_ sets d.y to 2, not 0.

The next two question relate to the following partial deﬁnition. In our GUI application we decide we need a widget that contains two equally spaced buttons and that this widget is to appear within the main window of the application under the label as shown below [see the exam paper].

class ButtonsFrame(Frame):

def \_\_init\_\_(self,parent):

Frame.\_\_init\_\_(self, parent.root)

b1 = Button(self, text= "A")

b2 = Button(self, text = "B")

## line 1 ##

class MainWindow(object):

def \_\_init\_\_(self, root):

self.root = root

Label(root, text="Buttons Below").pack()

bf = ButtonsFrame(self)

## line 2 ##

35. What is the required code for ## line 1 ##?

(a) b1.pack(); b2.pack()

(b) b1.pack(side=LEFT); b2.pack(side=LEFT)

(c) b1.pack(side=LEFT, expand=1); b2.pack(side=LEFT, expand=1)

(d) b1.pack(side=TOP, expand=1); b2.pack(side=TOP, expand=1)

(e) More than one of the above is correct.

Answer: c. We want to pack them left-to-right, and we want expand to make the space around the buttons within the frame.

36. What is the required code for ## line 2 ##?

(a) bf.pack()

(b) bf.pack(fill=X)

(c) bf.pack(expand=1)

(d) bf.pack(fill=X,expand=1)

(e) More than one of the above is correct.

Answer: d. We want the frame to fill the space it has, so that the Buttons can then get that extra space. We want to expand the frame to get it to have extra space.

37. Below is a recursive deﬁnition of a (non-empty) linked list class. A linked list consists of a head (a value) and a tail (a linked list).

class LinkedList:

def \_\_init\_\_(self, head, tail):

self.\_head = head

self.\_tail = tail

def head(self): return self.\_head

def tail(self): return self.\_tail

def isLast(self): return self.\_tail == None

def values(self):

result, scan = [], self

while not scan.isLast():

result.append(scan.head())

scan = scan.tail()

result.append(scan.head())

return result

Below is an incomplete recursive deﬁnition of concat for concatenating two linked lists to produce a linked list. As an example of its use consider the following involving the linked lists x and y.

>>> x.values()

[1, 2, 3, 4]

>>> y.values()

[5, 6, 7, 8]

>>> z = concat(x, y)

>>> z.values()

[1, 2, 3, 4, 5, 6, 7, 8]

What expression needs to replace #### in order to complete the following recursive deﬁnition of concat?

def concat(xs, ys):

if xs.isLast():

return LinkedList(xs.head(), ys)

else:

return ####

(a) LinkedList(ys.head(), concat(ys.tail(), xs))

(b) LinkedList(ys.tail(), concat(xs.head(), ys))

(c) LinkedList(xs.head(), concat(xs.tail(), ys))

(d) LinkedList(xs.tail(), concat(xs.head(), ys))

(e) None of the above

Answer: c. The first value in the concatenated linked list is the first value in xs (i.e. xs.head()). Everything else is the remaining elements of xs and all of ys (i.e. concat(xs.tail(), ys))

38. What is the time complexity, in terms of the length of the list, of the following function that reverses the elements of a list. You may assume the arithmetic operators used in the function and accessing and appending elements of a list are constant time operations. You may also assume the range function used is linear in the ﬁrst argument.

def reverse(xs):

"""Return the reverse of xs. """

result = []

for i in range(len(xs), 0, -1):

result.append(xs[i-1])

return result

(a) Constant

(b) Logarithmic

(c) Linear

(d) Quadratic

(e) Exponential

Answer: c. The call to range() is linear, and the loop is also linear. These two happen separately (not inside each other), so the resulting complexity is the sum of two linear things, which is linear.

39. What is the time complexity, in terms of the length of the list, of the following function that tests if repeats occur in a list. You may assume accessing elements of a list, calculating the length of a list and the arithmetic operators are constant time operations.

def repeats(xs):

"""Return True iff repeated elements occur in xs

repeats(list(X)) -> bool

"""

i = len(xs)

while i > 0:

i -= 1

j = i

while j > 0:

j -= 1

if xs[i] == xs[j]:

return True

return False

(a) Constant

(b) Logarithmic

(c) Linear

(d) Quadratic

(e) Exponential

Answer: d. We have two nested while loops. The first one goes from i=len(xs) down to 1, taking about len(xs) steps. The second one goes from j=i down to 1, which averages to about len(xs)/2 steps. The loops are nested, so the running time is the product of two linear things, which is quadratic.

40. What is the value of y after the following has been evaluated?

g = lambda x,y: (x, len(y))

f = lambda x,y: x > y+1

xs = [0,1,2,3]

y = [g(y, [x for x in xs if f(x,y)]) for y in xs]

(a) [(0, 0), (1, 1), (2, 2), (3, 3)]

(b) [(0, 3), (1, 2), (2, 1), (3, 0)]

(c) [(0, 2), (1, 1), (2, 0), (3, 0)]

(d) [(0, 0), (1, 2), (2, 1), (3, 0)]

Answer: c

- it is a list of pairs (since the return value of g is a pair)

- the first element of each pair is a number from xs, i.e. 0, 1, 2, 3

- the second element is the number of things in xs bigger than the first +1.

- two things in xs are > than 0+1, so 0 is paired with 2.

- one thing in xs is > than 1+1, so 1 is paired with 1. etc.