# CSSE1001: Sem. 2 2008 exam answers

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

## Get more out of your study time. Join UQAttic's Semester 1, 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.

### 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.

The following answers were initially made by a past tutor of this course, the original version is published at<http://codefisher.org/csse1001/> with the disclaimer: “I don't guarantee the answers are 100% correct, however as someone who achieved a 7 for the subject, used Python regularly ever since, and tutored it 3 times I would suggest the chance of an error is small.”

Feel free to modify, add to, or comment on these answers.

1. a)

2. c) it always rounds down, towards negative infinity.

3. d) integer can't be concatenated with strings, it has to be converted to one first

4. c) the string is converted to an int, then they are added.

5. a) the integer is converted to a string, then the two strings are concatenated

6. a) counting starts from 0, so the '.' is in the 1st position

7. d) starting at the second character, and going up to but not including the 6th. As a quick

mental check, the difference between the two numbers is the same as the length of the string.

8. c) start at the -4th character (the last is -1) and go to the end of the string, remember the

second number is not included, so the first two would leave the 2 off.

9. b) start the the 5th (last) character and go down to, but don't include the 1st character in steps of minus one.

10. d) it will split at the first four '0', empty strings are not removed

11. c) the pop will first remove and return the 2, then it will be inserted in position 3

12. d) square brackets on a non existent item always gives an error

13. d) if you evaluate all the get function, the last line becomes [1,2]+[3]+ None. You can't add

None to lists.

14. a) [1,2]+[3]+[] will work this time and give the expected result

15. e) the function takes only one argument, not two

16. c) this time it works, and since a < b, the result is (2, 6-2)

17. a) no simple way to do this, just evaluate the function twice.

18. a) the function simply copies the string

19. e) it will run over the end of the list, and trying to access something not in the list causes an

error

20. b) the loop will never run, as n is never less then m

21. c) it is joining the corresponding items in both lists together

22. c) since the first list in what controls the how many times the loop goes around, it will just

leave the last one of the second list.

23. e) the second list is two short, and it will throw an error trying to access the 2nd item in it.

24. e) not sure what the lectures answer would be, but the code needed to be d) plus a break

statement to stop the loop running since a solution was found. d) may well have been

marked as correct though. I think the question has errors in it, just ignore.

25. a) a line with 'Total:' was found, so stop the loop and let None be returned

26. b) Look at the loop invariant - r\*(t\*\*p) == x\*\*y - the trick to making the question easy, is

simply working out what keeps that true. The code is changing p so it becomes half of what

it was before. So that means that the value inside the brackets is the square root of what it

was before. To make it equivalent to what it was before, something needs to be squared.

27. c) the first argument regardless of name is the object – by convention called self. data

attributes, also called instance variable are prefixed with self, class variables are prefixed

with the class name.

28. d) it can do any all or none of the above.

29. c) the default one appears to compare memory addresses. Which is not very useful. The

function needs to be defined to give a reasonable definition of it. Of course you could also

be smart and say d) because you happen to know about the \_\_cmp\_\_ method.

30. e) f() calls g() which does not exist on A, so that gives an error.

31. c) \_\_init\_\_ sets self.x to 2. g() returns 2\*self.x and f() returns 2\*g(), so 2\*2\*2

32. c) g() + self.y is now 4 + 4

33. d) the f() method being called in the one on A, g() is now adding self.x and self.y so 2\*(2+4)

34. c) if you did the second assignment that should be clear

35. d) this question more tests your maths then your programming. 36. b) is the only one that does what is described. a) pops the first argument and passes that is,

which is quite wrong, and c) does not include the last element.

37. c) you want to insert x, the first element, into all the sub perms found

38. d) The loop will go around for each element in the loop in the while, then almost all again in

the for. So that gives something like (x^2)/2

39. b) though there are two loops, it only goes over each element once. -- Wouldn’t that make it linear, so c)?

40. c) the inner list comprehension turns it into a list of tuples that is make of the length of the word and the word. Then the outer filters out any that have a length that is not greater then 4.