CSCI 1100 Exam #1

Aaron Taylor

TOTAL POINTS

85 / 100

QUESTION 1

1 Question 1 10 / 10

- **0 Correct**
- 1 Part 1 error not correctly explained
- 1 Part 1 error incorrectly identified
- 1 Part 2 error not correctly explained
- 1 Part 2 error incorrectly identified
- 1 Part 3 error not correctly explained
- 1 Part 3 error incorrectly identified
- 1 Part 4 error incorrectly identified
- 1 Part 4 error not correctly explained
- 1 Part 5 error not correctly explained
- 1 Part 5 error incorrectly identified
- 0 Click here to replace this description.

QUESTION 2

2 Question 2 7 / 11

- 0 Correct
- 1 Does not import math (ok if uses **(0.5) instead)
- 2 Does not read input
- **2 Does not convert input to float**
- 1 Does not compute and print the first square root correctly
- 2 Does not compute and print the second square root value correctly
- **2 Does not print with one decimal place formatting**
- 1 Has additional spaces in formatting

convert your results into one decimal float before printing... incorrect formatting

use math.sqrt() instead of sqrt()

convert input into float before finding sqrt

QUESTION 3

3 Question 3 11 / 11

- **0 Correct**
- 2 Formatting is incorrect in the first print statement
- 2 Formatting is incorrect in second print statement
- 2 Formatting is incorrect in third print statement
- 2 Average is computed incorrectly (no floats or an error in the formula)
- 1 Average is printed to more than 1 decimal place
- 3 Does not compute the range correctly
- 3 Missing Line 1 in output
- 4 Missing Line 2 in output
- 4 Missing Line 3 in output

QUESTION 4

4 Question 4 5 / 10

- 0 Correct
- 2 Function is not defined properly (with 'def' and parameters)
- 2 Function returns something, but does not print
- 1 Does not print first two lines correctly
- 2 Does not print 5 lines of only one dot (missing one or extra newline)
- 1 Does not print 10 dots
- 1 Last line has additional space between dots and s3
- **1 Does not make string upper case**
- **2 Incorrect syntax, name error, type error, or similar interpreter error**
- **2 Arbitrary additional newline characters**
- 1 Additional exclamation mark
- 1 Does not print s3 at all
- 3.5 Hard-coded strings instead of using variable names

QUESTION 5

5 Question 5 10 / 10

- **0 Correct**

- 2 Does not read the inputs
- 2 Does not use replace correctly (string.replace call)
- 2 Does not replace the word with equal number of stars
- 1 Replacement is not case sensitive
- 1 Does not print the line "Censored sentence:"
- 2 Does not print the resulting sentence that is returned by string.replace call (partial credit if resulting sentence is, at the least, stored in a variable)

QUESTION 6

6 Question 6 10 / 12

- 0 Correct
- 1 Does not read user input
- 1 Does not convert input to integer
- **2 Does not compute the first digit correctly**
- **2 Does not compute the third digit correctly**
- 3 Does not correctly assemble the numbers to be multiplied
- 1 Final product is not correct
- 2 Final output is not correctly formatted
- **+ 2 Point adjustment**

> Partial Credit: Most of the idea is there. You can't do num[0] when num is an int though

QUESTION 7

7 Question 7 12 / 12

- **0 Correct**
- 2 Function is not defined correctly (def or parameter missing)
- 2 Function does not return a value
- 3 Function does not compute the calories correctly
- 1 Program does not read user input
- 2 Program does not call function correctly
- 2 Final output is formatted correctly
- 2 Syntax error
- 1 Incorrect indentation
- 3 Erroneous syntax errors
- 3 Multiple solutions submitted
- 3 Used loop

QUESTION 8

8 Question 8 11 / 12

- 0 Correct
- 2 Does not read three inputs
- 2 Does not convert inputs to integer or float
- 2 The condition in the if statement does not use ==
- 2 The condition in the if statement is incorrect
- 2 The if statement has extra incorrect conditions
- 2 The print statements for the if statement are incorrect
- 1 Does not have proper indentation
- **1 Has some incorrect syntax**

> Missing closing parenthesis after converting to int. Missing colon after if statement.

QUESTION 9

9 Question 9 9 / 12

- 0 Correct
- **1 Part A output is half correct**
- 3 Part A output is completely incorrect
- 1 Part A output is not formatted correctly
- 1 Part B output - first line is incorrect
- 1 Part B output - second line is incorrect
- 1 Part B output - third line is incorrect
- **1 Part B output - fourth line is incorrect**
- **1 Part C - cat is not printed 4 times**
- 1 Part C - cat is formatted incorrectly
- 1 Part C - fish is not printed 4 times
- 1 Part C - fish is formatted incorrectly
- 0 Click here to replace this description.

C

Computer Science 1 — CSci 1100
Exam 1
September 28, 2015

RCS ID: taylor5 @rpi.edu **Name:** Aaron Taylor

RIN # : 661537404

Circle your lab section

Sec. 1	T 10 (Low 3112), Jeramey
Sec. 2	T 10 (Walker 5113), Rahul
Sec. 3	T 12 (Sage 2715), Rahul
Sec. 4	W 10 (Sage 3101), Jeramey
Sec. 5	W 12 (J-Rowl 2C30), Dean
Sec. 6	W 12 (J-Rowl 2C06), Naveen
Sec. 7	W 12 (Eaton 215), Rhaad
Sec. 8	W 2 (Eaton 215), Rhaad

Sec. 9	W 2 (J-Rowl 2C14), Naveen
Sec. 10	W 2 (J-Rowl 2C22), Young
Sec. 12	W 2 (Sage 2704), Jassiem
Sec. 13	W 4 (Eaton 215), Jassiem
Sec. 14	W 4 (Lally 104), Young
Sec. 15	W 6 (Lally 102), Partha
Sec. 16	T 12 (Darrin 235), Partha

Problem	Points	Score
1	10	
2	11	
3	11	
4	10	
5	10	
6	12	
7	12	
8	12	
9	12	
Total	100	

Instructions:

- You have 90 minutes to complete this test.
- You may use only one double-sided crib sheet. Otherwise, put away all books, laptop computers, and electronic devices.
- Please read each question carefully several times before beginning to work.
- In all the questions, your output must match exactly the given output (do not insert additional spaces if they are not in the output).
- We generally will not answer questions except when there is a glaring mistake or ambiguity in the statement of a question.
- Except for problem 1, there are no Python syntax errors anywhere on this test.
- Unless otherwise stated, you may use any technique we have covered thus far in the semester to solve any problem.
- Please state clearly any assumptions that you have to make in interpreting a question.
- There are **9 questions** on this test.
- When you are finished with this test please turn it in to the proctor and show him or her your student id. You will then be free to leave the exam room.

Only use the space below for your answers, do not write on the back.

1. (**10 points total; 2 points each**) For each of the following Python code segments, determine if it uses correct syntax. If there is a syntax error, circle and explain the **first** syntax error.

```
x = 80
y = 'oak'
z = 'tree'
print x + ' ' + y, z + 's'
```

#can't add integers and strings

```
name = "Alice"
print 'Your name has' + len(name), + "letters."
```

#needs only a "+" or ",", not both
#strings and integers can't be added

```
full name = 'total supermoon eclipse'
full name.capitalize()
print full name
```

#no spaces in variables

```
cnt = raw_input('Enter a number => ')
print 'Hello!' * (cnt+cnt)
```

#would be a string not an integer, can't multiply strings

```
x = 5*y-6
y = 3
print x,y
```

#y is not defined before defining function x

2. (**11 points**) Write a program that reads an input containing a float, then prints the square root of the input number and the square root of the square root. Remember, the square root function comes from the `math` module.

   Here is an example output of the program:

```
Enter a number => 16
Sqrt(16.0) = 4.0
Sqrt(4.0) = 2.0
```

-input (float)
print √input)=

```
Import math
num = int (raw_input('Enter a number => '))
root = sqrt(num)
print "Sqrt(" + str(num)+") = ", root
root2= sqrt(root)
print "Sqrt(" + str(root) +") = ", root2
```

Only use the space below for your answers, do not write on the back.

3. (**11 points**) Suppose you are given four integer scores already stored in variables `s1,s2,s3,s4`.

   Write a program that uses these variables to compute and print out the values, their average, and their range computed as the difference between the maximum and the minimum value stored in these variables.

   For example, if you are given the following existing variables:

```
s1 = 2
s2 = 9
s3 = 6
s4 = 5
```

   Your program should print:

```
Values: 2, 9, 6 and 5
Average: 5.5
Range: 7
```

```
print 'Values:', str(s1)+",",str(s2)+",", s3, "and",s4
print "Average:",, ((s1+s2+s3+s4)/4.0)
range = max(s1,s2,s3,s4) - min(s1,s2,s3,s4)
print "Range:", range
```

4. (**10 points**) Write a function called `emphasize(s1,s2,s3)` that takes three strings as input. The function prints the first two strings each on a separate line without alteration, then prints five lines with a single dot (`'.'`), and then the final string in upper case on the last line, preceded by 10 dots.

   Here is an example run of your function:

```
>>> emphasize('This','is','Sparta!')
This
is
.
.
.
.
.
..........SPARTA!
```

```
def emphasize(s1, s2, s3):
    print s1
    print s2
    print '.\n' * 5
    print('.' * 10) + s3.upper()
```

5. (**10 points**) Write a piece of code that reads from the user first a sentence and then a word to be censored. Your code must replace all occurrences of the censor word with a number of stars equal to the length of the word, and then print `Censored sentence:` and then the resulting sentence (note that the replacement is case sensitive).

   Here is an example run of your program:

```
Sentence => enough is enough. I have had it with these monkey fighting snakes on this monday-to-friday plane.
Word => enough
Censored sentence:
****** is ******. I have had it with these monkey fighting snakes on this monday-to-friday plane.
```

```
sentence = raw_input("Sentence => ")
word = raw_input("Word => ")
new.sentence = sentence.replace(word, "*"*len(word))
print "Censored sentence: "
print newsentence
```

6. (**12 points**) Write a program that reads a three digit number and determines one number composed of the first and third digits of this number and another one composed of the third and first digits. It then prints both numbers and the result of multiplying them.

   Here is an example run of your program:

```
Number ==> 123
13 * 31 = 403
```

```
num = int(raw_input("Number ==> "))
num1 = str(num[0]) + str(num[2])
num2 = str(num[2]) + str(num[0])
product = int(num1) * int(num2)
print num1, "*", num2, "=", product
```

7. (**12 points**) Suppose a health tracking system like Fitbit stores your activity in a string like `'++-*****+-**'` where a plus sign (+) means you went up 10 stairs (consumes 3 calories), a minus sign (-) means you went down 10 stairs (consumes 2 calories) and a star (*) means you walked 100 steps (consumes 5 calories).

   Write a function called `find_cal(activity)` that takes as input a string like this one and returns the total calories you have consumed. Do not use loops.

   Now write a program that reads from the user an activity string and uses the function you wrote to compute and then print the total calories consumed.

   Here is an example run of your program:

```
Enter activity => ++-*****+-**
You consumed 43 calories
```

```
activity = raw_input("Enter activity => ")
def find_cal(activity)
    cal3 = (activity.count("+")) * 3
    cal2 = (activity.count("-")) * 2
    cal5 = (activity.count("*")) * 5
    total = cal3 + cal2 + cal5
    return total
print "You consumed", find_cal(activity), "calories"
```

8. (**12 points**) A Pythagorean triple is a set of three positive integers `a,b,c` that can make up the three sides of a right triangle, satisfying the following formula:

$$c^2 = a^2 + b^2$$

Write a program that reads three numbers and prints `Correct input` if the numbers satisfy the formula and `Incorrect input` if they do not.

Here is a possible run of this program:

```
a => 3
b => 4
c => 5
Correct input.
```

Here is another possible run of this program:

```
a => 3
b => 4
c => 6
Incorrect input.
```

```
a=int(raw_input("a => "))
b=int(raw_input("b => "))
c=int(raw_input("c => "))

if (a**2)+(b**2) != (c**2):
    print "Incorrect input."
else:
    print "Correct input."
```

9. (**12 points total; 4 points each**) Assume each of the following is an individual program. Write the **exact** output of these programs in the area provided. Show your work at each step in the scratch area provided for partial credit. Note: there are no syntax errors in the code provided in this question.

Part a

```
x = 5
y = x + 3
x = x + 1
z = y % x
w = float(y/x)
print 'x: %d, y: %d' % (x, y)
print 'z: %d, w: %.2f' % (z, w)
```

Output: x: 6, y: 8
z: 0, w: 0.75

Scratch area:

8
6 0.75
8) 6.00 = 0
w = 0.75

Part b

```
def func(x):
    print '-'*x, x**2

x=2
func(3)
print x
print func(4)
```

Output: - - - , 9
2
- - - - , 16

Scratch area:

Part c

```
def mystring(s):
    return s + '\n'
def yourstring(s):
    return s * len(s)

print yourstring(mystring('cat'))
print mystring(yourstring('fish'))
```

(cat + '\n') * 3
(fish * 4) + '\n'

Output: cat
cat
cat
fish fish fish fish

Scratch area: