

Computer Science 1 — CSci 1100 — Spring 2016

Exam 1

February 22, 2016

RCS ID: _____ @rpi.edu

Name: _____

RIN # : _____

Circle your lab section

Sec. 1	T 10 (Sage 2704), Jeramey
Sec. 2	T 12 (Sage 2112), Simon
Sec. 3	T 2 (Sage 3101), Jeramey
Sec. 4	W 10 (Eaton 216), Simon
Sec. 5	W 12 (Eaton 215), Antwane
Sec. 6	W 2 (Sage 2704), Antwane
Sec. 7	W 10 (Eaton 215), Simon
Sec. 8	W 12 (Eaton 216), Antwane

Problem	Points	Score
1	15	
2	10	
3	12	
4	12	
5	12	
6	12	
7	12	
8	15	
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 2, 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 **8 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.

1. (15 points total; 5 points each) Assume each of the following is an individual program. Write the **exact** output of these programs in the area provided. Use the scratch area as you wish. There are **no syntax errors** in the code provided in this question.

Part a:	Answer:	Scratch area:
<pre> x = '4' print(x) x += '1' print(x) y = 2**3 print(y) y -= 3 print(y) x *= y print(x) </pre>		

Part b:	Answer:	Scratch area:
<pre> s = 'fun' + 'ny' print(s.replace('nn','N')) s = 'cart ' + ' wheel' print(s.replace('tw', 'X')) t = '''hello """"\nworld''' print(len(t)) print(t) </pre>		

Part c:	Answer:	Scratch area:
<pre> c = 'higher' def f1(a,b): if len(a) > len(b): print('a') return len(a) else: print(b) return len(a) + len(b) def f2(a): x = f1(c,a) return c + str(x) print(f1('bike', 'car')) print(f2('sheldon')) </pre>		

2. (10 points total; 2 points each) Assume each of the following is a separate program. For each program find the **first syntax** error, or decide there is no error. If there is an error, write in the solution box that there is an error, write the line number **and** very briefly describe the error. If there is no syntax error write **No Error**.

Part a:	Answer:
<pre>def f123_45(x): #1 print(x) #2 return len(x) #3 print(f123_45('678')) #4</pre>	
Part b:	Answer:
<pre>y = 6 #1 x += 45 #2 print(x) #3 print(x-7) #4</pre>	
Part c:	Answer:
<pre>def area(r,pi): #1 return pi * r * 2 #2 print(area(2.5, 3.14 * 2)) #3</pre>	
Part d:	Answer:
<pre>def volume(r): #1 print(r) #2 return 4/3*math.pi*r**3 #3 print(volume(5)) #4</pre>	
Part e:	Answer:
<pre>a = 6 #1 b = 73 #2 a = a<b #3 print(a) #4</pre>	

Write your answers in the box below only. Do not write on the back or outside the box.

3. (12 points) Write a Python program that asks the user for a sentence and then does the following. It prints the sentence with the first letter capitalized. Then it prints the sentence with every word capitalized. Next it prints the sentence in all lower case. Finally, it prints the length of the sentence, and it prints the number of times the letter "e" or "E" appears.

Here is one example

```
Enter a sentence => tHis iS a ranDOM fOUR e SenTEnce.  
Capitalized: This is a random four e sentence.  
Every word capital: This Is A Random Four E Sentence.  
Lower: this is a random four e sentence.  
Length: 33  
Times e appears: 4
```

Write your answers in the box below only. Do not write on the back or outside the box.

4. (12 points) Assume you have a right triangle with sides **a**, **b** and hypotenuse **c**. Write a Python program that asks the user for the integer length of **a** and of **b**. Then it prints the minimum of the values of the two sides, the maximum of the values of the two side, and the length of the hypotenuse **c**. Remember that **c** is the square root of **a** squared plus **b** squared. The value of **c** should be a float value and should be printed with one digit to the right of the decimal place.

Here is one example

```
Enter side a => 2
Enter side b => 1
Minimum of a and b is 1
Maximum of a and b is 2
The length of hypotenuse c is 2.2
```

Write your answers in the box below only. Do not write on the back or outside the box.

5. (12 points) Write a section of Python code that starts by reading a single, positive digit greater than 1 from the `raw_input` and then storing it in a variable `n`. Assume the input is correct. The code should output a string that shows `n` being multiplied by itself `n` times and then outputs the correct value of `n**n`. Here is an example of what the output might look like when run as a program in the Wing IDE

```
Enter a digit greater than 1 => 4
4*4*4*4 = 256
```

Write your answers in the box below only. Do not write on the back or outside the box.

6. (12 points) Write a Python function called `remove_shorter` that takes two strings as arguments, decides which string is longer, and returns a string that has all occurrences of the shorter string removed from the longer. If the strings are the same length the function should return the concatenation of the two strings. Here are three examples

```
>>> print(remove_shorter('cheese shops have no cheese', 'che'))
ese shops have no ese
>>> print(remove_shorter('cat', 'cat and rat came back'))
and rat came back
>>> print(remove_shorter('spanish', 'inquire'))
spanishinquire
```

Write your answers in the box below only. Do not write on the back or outside the box.

7. (12 points) Write a Python program that requests two strings from the user using `raw_input()`. The program should then print each string on a separate line such that the second string starts just below the middle of the first string, followed by a line of asterisks that underline both strings and ends with the end of the second string.

Here is one example with an even-length first string

```
Enter string1 => flying
Enter string2 => circus
```

```
flying
  circus
*****
```

Here is a second example with an odd-length first string

```
Enter string1 => bye
Enter string2 => good
```

```
bye
  good
*****
```

Write your answers in the box below only. Do not write on the back or outside the box.

8. (15 points) This question is in two parts. For part (a), write a function called `prod_digits` that takes as an argument a three-digit integer and returns the product of the digits. Here is an illustration of how it should run:

```
>>> prod_digits(252)
20
>>> prod_digits(631)
18
```

For part (b) write a program that asks the user for two three digit numbers, calls `prod_digits` from part (a) to calculate the product of the digits in each two digit number, and outputs the sum of the two products. Here is an example of running the program.

```
Enter a three digit number => 252
Enter another three digit number => 631
Sum is 38
```
