CS 61A Structure and Interpretation of Computer Programs
Summer 2017

Quiz 1 Solutions

INSTRUCTIONS

- You have 10 minutes to complete this quiz.
- The exam is closed book, closed notes, closed computer, closed calculator.
- The final score for this quiz will be assigned based on **effort** rather than correctness.
- Mark your answers **on the exam itself**. We will *not* grade answers written on scratch paper.
- For multiple choice questions,
  - ☐ means mark **all options** that apply
  - ◯ means mark a **single choice**

Last name	
First name	
Student ID number	
CalCentral email (`_@berkeley.edu`)	
Teaching Assistant	◯ Alex Stennet ◯ Kelly Chen ◯ Angela Kwon ◯ Michael Gibbes ◯ Ashley Chien ◯ Michelle Hwang ◯ Joyce Luong ◯ Mitas Ray ◯ Karthik Bharathala ◯ Rocky Duan ◯ Kavi Gupta ◯ Samantha Wong
Name of the person to your left	
Name of the person to your right	
All the work on this exam is my own. **(please sign)**	

1. (5 points) Proceed with call-tion

For each of the expressions in the table, fill in the bubble corresponding to the output displayed by the interactive Python interpreter when the expression is evaluated. If an error occurs, choose "Error". If a function value is displayed at any time during evaluation, choose "Function". If the output is not any one of the given responses, choose "Other". The first two rows have been provided as examples.

Recall: The interactive interpreter displays the value of a successfully evaluated expression, unless it is `None`.

Assume that you have started `python3` and executed the following statements:

```
from operator import mul

x = 3

def square(x):
    return mul(x, mul(x, 1))

def cube(x):
    x = x + 1
    return print(square(x) * x)
```

Expression	Interactive Output		
`pow(2, 3)`	`8`		
`print(4, 5) + 1`	`4 5` `Error`		
`square(2) + square(x)`	○ `2` ○ `3` ○ `4`	○ `9` ○ `4 + 9` ● `13`	○ Function ○ Error ○ Other
`cube(3)`	○ `27` ○ `27 None` ○ `27` `None` ● `64`	○ `64 None` ○ `64` `None` ○ `square(3) * 3` ○ `square(4) * 4`	○ Function ○ Error ○ Other

`print(square(3), print(square(4)))`	○ `9 16`
	○ `9`
	`16 None`
	○ `9 None`
	`16`
	● `16`
	`9 None`
		○ `16 None`
		`9`
		○ `9 None`
		`16 None`
		○ `square(3)`
		`square(4)`
		○ `square(3)`
		`square(4) None`
			○ `square(3)`
			`square(4)`
			`None`
			○ Function
			○ Error
			○ Other

`print(cube(square(2)))`	○ `64` ○ `64 None` ○ `64` `None` ○ `125`	○ `125 None` ● `125` `None` ○ `cube(square(2))` ○ `cube(square(2))` `None`	○ Function ○ Error ○ Other