CSCI 150: Exam 1 Practice

September 12, 2023

Directions: Exam #1 occurs in two parts, an in-class and take-home.

In-class: The in-class portion will take place during regular class time on Monday, September 18. You will have 50 minutes to complete the exam. For the in-class portion, you may not use a computer, calculator, or other device. It will consist of evaluating the truth value of four boolean statements, tracing the execution of two pieces of code, and writing two straightforward functions by hand. You must work the in-class without reference to notes or any other resource.

Take-home: The take-home will be given to you as you leave the in-class exam. You should create a single Kaggle notebook, and upload your file via CodeGrade. You will be asked to write 4 separate functions, and your solutions should each be placed in a separate cell. Your solutions are due at the start of class on **Wednesday, September 20**. Like on the Codingbat conditional homework and the practice exam below, example input and output will be provided. For the take home, the *only* resources you may use are the following:

- any code you have created for class, including homework and labs
- any notes you have taken during class
- any information, code, or other material directly linked from the class homepage, `https://hendrix-cs.github.io/csci150/`
- anything in the official Python documentation, `https://docs.python.org/3/`

You may not talk to a classmate, friend (real-life or Facebook), Siri, ChatGPT, or anyone other than me about this exam until you turn it in, nor search the Internet or library or any reference other than those listed above for assistance. You may not even mention anything about how long it took you to complete the exam, that you found problem #2 particularly difficult (or easy), or in fact talk at all about the exam or Computer Science with anyone other than your instructor from Monday, 8:10am–Wednesday 12:00pm. Anyone who *gives* answers is equally in violation of the Academic Integrity Policy as one who *receives* them.

If you have ANY questions about whether something you would like to do is allowed, please ASK!

Specification: Exam #1 will be marked **Complete** provided:

- Each problem is attempted
- On the in-class:
  - 3 out of 4 boolean problems are correct, with the 4th having at most one mistake (this means that enough work must be shown so that I can tell!)
  - At least one tracing problem is completely correct, and the other has a single mistake
  - Excepting minor syntax issues, at least one by-hand code writing problem is correct, and the other misses a single case; both answers exhibit clear evidence that the student understands the basic structure of Python functions and conditionals.
- On the take-home:
  - No syntax errors in any code
  - No runtime errors with any of the examples given
  - 3 out of 4 produce the right answer in all circumstances
  - the 4th produces the right answer for the given examples, but may have other errors

Exam #1 will be marked **partially complete** provided:

- At most one problem on the in-class and one on the take-home are not attempted
- On the in-class:
  - At least 1 boolean is completely correct, and one other shows work which indicates at most one mistake
  - Tracing: All variables outside of conditionals are properly set up, and at least three are updated correctly
  - By-hand code: At least one attempted solution would work in some circumstances; syntax errors are minor, and the work shown shows a basic grasp of Python structure
- On the take-home:
  - No syntax errors in any code
  - At least 3 functions produce no runtime errors with the examples given and 2 produce no runtime errors at all

– At least 1 function produces the right answers in all circumstances.

The instructor will return your exam along with a rubric detailing your progress and assessment. If you earned less than a **Complete**, instructions will be provided on that rubric about how to attempt to raise your assessment. In most cases, an **Incomplete** can only be "upgraded" to a **Partially Complete**, though the instructor reserves to right to allow **Missing** to become **Complete** in some circumstances.

In general, the student who seeks an upgrade will be expected to correct all errors, complete additional problems as assigned, and meet with the instructor in office hours to discus their work.

In-Class Practice

1. You are given

   ```
   a = 4
   b = 3
   c = 'test'
   d = 'fun'
   ```

   Evaluate the truth value of each Python expression below:

   - `False or b > a`

   - `(True and False) or (not False or False)`

   - `(a - b) > 1`

   - `((a * b) - 12) == 0`

   - `(not (c > d)) and (a > b)`

2. Trace the execution of the following Python code

   Showing your work (*e.g.* evaluation of expressions) is not required, but makes it much easier to give partial credit if you make a mistake.

```
a = 5
b = 8
z = b - a
a = a + 9

if a == a + 9:
    b = 2
    z = 3
    a = 111

c = z + b
```

3. Trace the execution of the following Python code

   Showing your work (*e.g.* evaluation of expressions) is not required, but makes it much easier to give partial credit if you make a mistake.

```
i  = 3
animal  = 'cat'

if i < 1 or (animal == 'cat' and i > 1):
    i = i + 1
    animal = 'dog'
    first  = 'yay'

elif animal == 'dog' and i > 1:
    i = i + 1
    animal = 'pig'
    first  = 'boo'

if i > 5:
    i = i * 3
    second  = animal

else:
    i = i - 2
    second  = 'no'
```

For the next two, write code which will accomplish the following task. As practice for the in-class part of the exam, you should do this without any use of a Python interpreter – that is, you should do this entirely by hand.

4. Write a function `bigger` which takes in two integer parameters, `a` and `b`. It returns the larger of the two – in the case that they are equal, it simply returns that value. For example,

   `bigger(2,4)` will return `4`

   `bigger(10, -2)` will return `10`

   `bigger(7, 7)` will return `7`

5. Write a function `nice_outside` which takes in two parameters, an integer `temp` for the temperature, and a boolean `is_sunny`, which is true when the sun is out. The function returns a boolean by the following rule: It is nice outside (i.e. returns `True`) when it is sunny and the temperature is between 60 and 80 (inclusive) or when it is not sunny and the temperature is between 70 and 85 (also inclusive). For example

   `nice_outside(50,True)` will return `False`

   `nice_outside(74, True)` will return `True`

   `nice_outside(65,False)` will return `False`

   `nice_outside(83, False)` will return `True`

Take-Home Practice

1. Two trains a and b are on a collision course heading down the same track. If you know the speed of the two trains and how far apart they are, you can calculate when they will collide by the following formula:

$$collision = \frac{distance}{speed_a + speed_b}.$$

   Write a Python function, called `coll_time` that has three parameters, `d` which is the distance between them and `va` and `vb` which is are the speeds of the two trains. These should all be floats. This function should calculate how long before the trains collide using the above formula, and return the result.

   For example, you code should produce:

   `coll_time(100.0, 25.0, 75.0)` should return `1.0`

   `coll_time(50.0, 20.0, 5.0)` should return `2.0`

   `coll_time(100.0, 37.0, 25.0)` should return `1.6129032258064515`

2. Hal wants to write a function, called `alarm` that will tell him when to set his alarm clock. This function has two parameters, an integer variable for `day` according to 0=Sun, 1=Mon, 2=Tue, ..., 6=Sat, and a boolean variable `vacation` indicating if he is on vacation.

   The function should return a string of the form `7:00` indicating when the alarm clock should ring. Normally, on weekdays (Monday through Friday) the alarm should return `7:00` and on the weekend (Saturday or Sunday) it should return `10:00`. Unless he is on vacation—then on weekdays it should return `10:00` and weekends it should return `off`.

   For example:

   `alarm(3,False)` should return `'7:00'`

   `alarm(0,False)` should return `'10:00'`

   `alarm(4,True)` should return `'10:00'`

   `alarm(6,True)` should return `'off'`

3. European and American shoe sizes differ by a standard amount. You can approximate the European shoe size by using the following formulas.

   For men:

$$euroSize = \frac{9}{7} americanSize + 30.5$$

   For women:

$$euroSize = \frac{9}{7} americanSize + 29$$

Write a function `size` that takes two parameters, a string `gender` which has value `'M'` for men and `'F'` for women and a float `s` for the American size and then returns the European shoe size using the above formula.

For example:

`size('M',11.5)` should return `45.285714285714285`

`size('F',11.5)` should return `43.785714285714285`

`size('F',6.0)` should return `36.714285714285715`