Quiz 1

To get credit for this quiz, use the Quiz tool at eee.uci.edu to enter your answers, within the Sunday-to-Tuesday quiz period.

You’ll notice that some of the questions contain short reminders of what certain Python features mean. We won’t always include those, but this week you haven’t had much time to learn those meanings.

Problem 1 (7 points) Topic: Course policies and advice

Each of the following statements claims to be a policy, procedure, good advice, or other characteristic of ICS 31, but each is inaccurate, misguided, or wrongheaded in some way. Please change the statement (as little as necessary) to make it an accurate statement on the same topic.

(a) We expect students in ICS 31 to have at least two years’ experience writing software.
   
   **No previous programming experience is expected of ICS 31 students.**

(b) The best and fastest way to get answers to questions about the course material is to send Email to the instructor and TAs.

   *Posting on Piazza.com is best for questions about course material. Email is better for private correspondence.*

(c) In pair programming, two programmers split up the work so they can finish twice as fast.
   
   **In pair programming, two programmers work together, one “driving” at the keyboard and the other “navigating.” (In fact, studies show that this arrangement gets the work done faster.) Splitting up the work is not pair programming and is not how labs in ICS 31 should be done.**

(d) In pair programming, it’s best to find the most experienced partner you can, so your partner can do all the hard parts.

   **If you let your partner carry the weight, you won’t learn the material; it’s like asking someone else to exercise for you. Both partners need to discuss and understand the solutions. That can happen when one partner is much more experienced than the other, but it’s more of a give-and-take if the partners are more evenly matched.**

(e) This is college; attending class and lab are optional, so missing them has no bad consequences.

   **Lab is where you do your lab work with your partner each week, and where the TA and lab tutors are available to give you help. Lecture is where topics are introduced and explained, and where broader issues are covered; even with the UCI Replay recordings, you miss everything that’s not on the computer screen or the audio.**

(f) If you find some course concepts difficult or confusing, or if you see people in the lab who work a lot faster than you do, you should give up and take another class.

   **We don’t expect everything to be perfectly clear and easy right away, and neither should you: Ask your TA or instructor or lab tutor or Piazza for help. To succeed in this class and in ICS, you do not have to keep up with the biggest hot-shots.**

(g) Save your questions in lecture for the last five minutes of class.

   **We prefer that you ask questions as they occur to you and leave it to us to decide how and when to answer.**
Problem 2 (4 points) **Topic: Evaluating numeric and string expressions**

What is the value of each of the Python expressions below? (In other words, what would Python display if you printed each expression out?) Use these variables where appropriate:

```python
r = 17
TAX_RATE = 0.1
s = 'downtown'
```

(a) \( r \times \text{TAX\_RATE} \)

1.7

(b) \( \text{len}(s) + 10 \)  
   # The function \text{len}() returns the number of items in a sequence

18. (\( s \) has 8 characters, even though zero-based indexing numbers them 0 through 7.)

(c) \( (r + 5) \times (r - 7) \)

220

(d) \( s[2] \)  
   # Remember zero-based indexing

w  
(For a question like this, it’s okay if you said ‘w’, but a print statement wouldn’t display the quotation marks. On the other hand, if you typed \text{s[2]} in the interpreter—-the Python Shell—-it WOULD display the quoted value.)

(e) \( s[-1] + s[1] \)  
   # Negative index numbers count from the end

no  
(It may seem strange that \text{s[-1]} is the last character while \text{s[1]} is the second character. But a negative index counts backwards from the LENGTH of the string, 8 in this case, and 8-1 is 7 so \text{s[-1]} is \text{s[7]}, the last character.)

(f) \( s[0:2] \times 4 \)  
   # A colon in an index indicates a “slice”

dodododo

(g) \( s . \text{count}('o') \)  
   # count() returns how many times its argument occurs in the object

2  
(The character ‘o’ occurs twice in \( s \))

(h) \( 'own' \text{ in } s \)  
   # The operator “in” checks inclusion

True  
(The string ‘own’ occurs in \( s \). The “in” operator returns a boolean, True or False.)
Problem 3 (3 points)  

**Topic: Evaluating list and boolean expressions**

What is the value of each of the Python expressions below? (In other words, what would Python display if you printed each expression out?) Use this variable where appropriate:

\[ L = [2, 3, 5, 7, 11, 13, 17] \]

(a) \( L[-1] \)  
17

(b) \( L[0] + L[1] == L[2] \)  
True

(c) \( L[-1] - L[-2] <= L[1] \)  
False (17-13 <= 3, which is False)

(d) 15 in L  
False

(e) \( L[1:4] \)  
\[3, 5, 7\] (The square brackets and commas are important here; they're part of the Python syntax for lists)

(f) \( len(L + [19]) \)  
8 (There are 8 items in [2, 3, 5, 7, 11, 13, 17, 19])

Problem 4 (4 points)  

**Topic: Expressions and types**

For each of the parts of Problem 2—(a–h)—and Problem 3—(a–f)—say what type the value is. Choose from `int`, `float`, `bool`, `str`, or `list`.

2a: float, 1b: int, 1c: int, 1d: str, 1e: str, 1f: str, 1g: int, 1h: bool.

3a:int,2b:bool,2c:bool,2d:bool,2e:list,2f:int.
Problem 5  (7 points)  **Topic: Input and output**

What appears in the Python Shell window if you run the following code and the user enters *Sam Smith* and then *19* at the two points where the code expects input?

```python
print("Hello.  What’s your name?")
user = input()
print("Hello, ", user, ", how old are you?")
age = int(input())
print("Next year you will be", age + 1, "years old.")
```

Hello.  What’s your name?
Sam Smith
Hello, Sam Smith, how old are you?
19

Next year you will be 20 years old.
(At this point in the course, you don’t have to be precise about the space after “Sam Smith”, but you should show the two commas printed on that line because there are two commas within the quoted string constants being printed. It’s also okay for this question if you didn’t show the lines that the user typed (“Sam Smith” and “19”).)

Problem 6  (12 points)  **Topic: More course policies.**

All of these topics are addressed on the syllabus and linked pages.  *Go back and read them if you need to.*

(a)  Why is it important to read each assignment specification more than once?
Computing assignments, like all technical documents, contain a great number of technical details. It’s impossible to remember them all from a single reading.

(b)  When is the first midterm scheduled (or, if you don’t remember, where can you find out)?
The first midterm will be in lecture on Tuesday, January 26 (see the “Course structure” section of the syllabus).

(c)  Is it ever okay to e-mail the code you’ve written for an ICS 31 lab assignment to another student?
*It’s never okay to send code to a student who isn’t your partner on a given lab. Normally, you’ll be working with your partner in person, so the only reason to mail your partner some code is at the end of the lab section, so both partners have a copy of the most up-to-date version of their work. It would also be okay after the course is over.*

(d)  If you have a question on how your work was scored, whom should you ask first?
Your TA.

(e)  What do you do if you haven’t finished the whole lab assignment by the end of the day on Friday, when it’s due?
*Turn in whatever you have on the due date via Checkmate. If you’d like to work further, check with your TA on whether a late submission will get any credit (though it might be worth doing it even if there is no credit available).*

(f)  If you and your partner are running out of time, is it okay to split up the remaining work (you do some, your partner does the rest) and combine that work to submit it?
*No, that’s not pair programming. Pair programming means two people working together on the same task at the same time, one “driving” and one “navigating.” On top of missing the point and losing the benefits, separate work can be a gateway to messy academic honesty situations where someone gets in trouble without even realizing it because his or her partner submitted some copied work.*