

Name: (as it would appear on official course roster)	
Umail address:	@umail.ucsb.edu
Optional: name you wish to be called if different from name above.	
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")	

1

h02

CS8 M18

h02: Perkovic 2.4-2.5 (Objects, Classes, types, libraries), 3.1-3.2 (print, input, if/else, for), 3.3-3.5 (Functions), 4.1-4.2 (Strings, Formatting)

ready?	assigned	due	points
true	Thu 08/09 09:30AM	Thu 08/16 09:30AM	370

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE.

There is NO MAKEUP for missed assignments, and you may not submit work in advance, or on behalf of another person.

In place of that, we drop the lowest score (if you have a zero, that is the lowest score.)

READING ASSIGNMENT

Please read Perkovic 2.4-2.5 (Objects, Classes, types, libraries). Then complete these problems.

- (10 pts) Please fill in the information at the top of this homework sheet, including your name and umail address. If the other two items apply, please fill them in as well. Please do this every single time you submit homework for this class. It is important to fill in both name and umail every time, since handwriting is sometimes difficult to decipher. Having both helps us ensure you get credit for your work.

Also: JUST write your name at the top of all sheets and turn in all sheets UNCONNECTED.

DO NOT staple, paper clip, spit-fold-and-tear, or do ANYTHING that would make it difficult to automatically feed your paper through a scanner.

- As discussed in Section 2.4, the `type()` function returns the type of a Python value. When you pass a variable such as `x`, `type(x)` returns the type of the value that the variable `x` currently refers to.

Assume that the following assignment statement has been executed:

```
schools=["UCSB","Stanford","UCSD","Cal Poly"]
```

What will each of the expressions below evaluate to? As a reminder, strictly speaking, Python will print types in the format `<class 'int'>`, `<class 'float'>`, `<class 'str'>`, etc. so please use exactly that format for full credit.

Points	Expression	Result	Points	Expression	Result
(4 pts)	<code>type(3)</code>		(4 pts)	<code>type(1+2.5)</code>	
(4 pts)	<code>type('3')</code>		(4 pts)	<code>type(2 * "3")</code>	
(4 pts)	<code>type("3.5")</code>		(4 pts)	<code>type((3,3))</code>	
(4 pts)	<code>type(3.5)</code>		(4 pts)	<code>type(schools)</code>	
(4 pts)	<code>type([3,5])</code>		(4 pts)	<code>type(schools[0])</code>	

2

h02

CS8 M18

3. (10 pts) Assume that `cases` is the name of a variable with a `float` value that you want to convert to an integer value. Write a Python expression that converts `cases` to an integer (throwing away any fractional part)
4. (10 pts) Assume that `courseNum` is an integer that represents the numeric part of a course number (e.g. 3, 8, 130, 16, 24). Write a Python expression that converts `courseNum` to a string (i.e. `<class 'str'>` in Python)
5. (10 pts) If you want to check whether `x` is greater than 10, and `y` is greater than 5, you can write the Python expression

```
(x > 10) and (y > 5)
```

If `x` has the value 20, and `y` has the value 17, this evaluates to `True`.

But what if we accidentally wrote it as:

```
(x > 10) + (y > 5)
```

What would this expression evaluate to, assuming the same values for `x` and `y`? (The answer requires you to read the section carefully, and then apply what you have learned. I suggest you try that first before trying it at the Python command line.)

6. Both Sections 2.4 and 2.5 mention a type of function that is called a *constructor*.
- (5 pts) In general, what do constructors do? (Don't give an answer that is specific, for instance, to only the constructor for the `int` data type).
 - (5 pts) *Before* using the constructor for a `Fraction` object, a particular line of Python code must be written. What is that line of code? (Be very careful about spelling and upper vs. lower case.)
 - (10 pts) Assuming that line of code has been typed (the one mentioned in the previous question), how do you create a `Fraction` object that holds the fraction $\frac{4}{5}$ (i.e., “four fifths”), and makes the variable `ratio` refer to that object?

READING ASSIGNMENT Please read Perkovic 3.1-3.2 (print, input, if/else, for). Then complete these problems.

7. (10 pts) Section 3.1 discusses a feature of IDLE called “restarting the shell”. What does the book say happens when you restart the shell?

8. (10 pts) Section 3.1 also discusses the `input()` function, which is used to get input from the user. It indicates that the result of the `input()` function always comes back as a particular data type, no matter what the user types in. What is this data type?

9. Section 3.2 discusses the fact that you can *nest* control structures in Python.

a. (10 pts) What does it mean to *nest* control structures in Python?

(Note: you won’t find the answer spelled out in the text. It is implied, but not clearly explained. So you’ll have to do some thinking to figure out how to answer. That’s on purpose.)

b. (10 pts) What can you see in the Python code (or do you have to write in your own Python code) to indicate that something is “nested”? (This is also in the text, but implied rather than stated.)

3
h02
CS8 M18

4

h02

CS8 M18

10. (10 pts) Section 3.2 discusses iteration, which means doing things over and over again, in a loop. As we will learn, there is more than one kind of loop in Python, but there is a particular kind of loop discussed in Section 3.2 that allows us to process each character in a string, or each item in a list, one at a time.

What is the Python word that starts that kind of loop?

11. (10 pts) On p. 56-57, the textbook suggests using the `eval()` function to convert the result of `input()` from the user if it going to be used as something other than a string (e.g. as a number, list, etc.)

THIS IS A POTENTIALLY DANGEROUS HABIT TO GET INTO. DON'T DO IT.

(We'll discuss this in more detail in lecture, but the short version is: this is the kind of habit that leaves security vulnerabilities open for hackers. There isn't any real danger in an intro course, because the software you write won't be run by anyone except you and your TA, but in general, this is VERY bad habit to get into.)

So what can we do instead? Well, you actually already saw one option—a technique that was discussed in Section 2.4. Suppose we have this Python statement:

```
ageAsString = input('Please enter your age in years: ')
```

Using techniques from Section 2.4, but NOT using `eval`, what line of Python code can you write to convert the value `ageAsString` into an `int` value?

12. (20 pts) Read about the `if` and `else` statements in Section 3.2, including what happens when the various programs in Section 3.2 are run.

(If you want to run them yourself to see what happens, I strongly encourage that! And: it is ok to type them in and run them, even including the `eval` function calls; I don't want to scare you too much about that. The `eval` function is only risky when someone you don't trust is putting input into your code, e.g. when you are making an app that the general public will use.)

Once you think you understand how `if` and `else` works, consider this code, which is incomplete. Add the code that is needed so that if the user puts in something other than `CA` for the state, the code will print "Out of State Tuition". Be sure to indent your code properly, and include all needed punctuation.

```
homeState = input("Please enter your state: ")

if homeState=="CA":
    print("In State Tuition")
```

13. (10 pts) Section 3.2 discusses how to use the `range` function with a loop so that you can cause a variable to take on a sequence of values—the sample code illustrates this by printing those values. (Later we'll do other things with those sequences of values such as adding them, or multiplying them, or using them to index into a list or string, or to draw things.)

In the space below, write a few lines of Python that uses a loop with the `range` function to print the values shown in the box at right. (Your answer must use a loop with `range` to get credit.)

```
3
5
7
9
```

READING ASSIGNMENT Please read Perkovic 3.3-3.5 (Functions). Then complete these problems.

14. (20 pts) Page 68 lists this Python function definition that corresponds to the mathematical function $f(x) = x^2 + 1$

1	<code>def f(x):</code>
2	<code> res = x**2 + 1</code>
3	<code> return res</code>

Note that we can also write this more simply without the `res` variable, saving a line of code.

1	<code>def f(x):</code>
2	<code> return x**2 + 1</code>

Which version is better? That's a matter of *programming style*. Often, but not always, writing in fewer lines of code is preferred. With experience, you'll gain a sense of when to do one vs. the other. For this problem, it is up to you which option to use.

In the space below, please write a function definition for the mathematical function $g(x) = 2x^2 - 3x + 4$

5
h02
CS8 M18

15. The following is a *function definition*. Note that `def` is an abbreviation for definition:

1	def f (x):
2	return x**2 + 1

Contrast this with a function call such as `f(3)`, `f(2+2)`, or `f(0.5)`. Those evaluate to `'10'`, `'17'`, and `'1.25'` respectively. Each of the problems below shows a function definition at left, then several function calls at right. For each function call, write what the function call evaluates to.

a.

<pre>def foo(x): return 1 + x * 2</pre>	pts	function call	write result here
	(5 pts)	foo(3)	
	(5 pts)	foo(-1)	

b.

<pre>def bar(s): return s[1:3]</pre>	pts	function call	write result here
	(5 pts)	bar("blue")	
	(5 pts)	bar("green")	

c.

<pre>def blerg(thing): return thing[-1]</pre>	pts	function call	write result here
	(5 pts)	blerg("blue")	
	(5 pts)	blerg([1,2,3])	
	(5 pts)	blerg(["UCSB", "Stanford", "UCLA"])	

d.

<pre>def fleek(a,b): return 2 * min(a,b)</pre>	pts	function call	write result here
	(5 pts)	fleek(3,5)	
	(5 pts)	fleek(7,1)	
	(5 pts)	fleek("UCSB", "UCLA")	

16. (10 pts) According to Chapter 3, what is a docstring?

17. (10 pts) Add an appropriate docstring to the Python function below that converts inches to centimeters.

```
def in2cm(inches):
    return inches * 2.54
```

6

h02

CS8 M18

READING ASSIGNMENT Please read Perkovic 4.1-4.2 (Strings, Formatting). Then complete these problems.

18. Section 4.1 discusses indexing and slicing of strings, as well as some methods (functions) that can be invoked on strings.

Assume that the following assignment statements have been executed.

```
fname="Chris"  
lname="Gaucha"  
schools=["UCSB","Stanford","UCSD","Cal Poly"]
```

What will each of the expressions below evaluate to?

Points	Expression	Result	Points	Expression	Result
(5 pts)	fname[0]		(5 pts)	lname[:2]	
(5 pts)	lname[0:2]		(5 pts)	lname[-3:]	
(5 pts)	schools[0]		(5 pts)	schools[1:]	
(5 pts)	schools[0:2]		(5 pts)	fname.find('h')	
(5 pts)	schools[0][0]		(5 pts)	lname.find('ch')	
(5 pts)	schools[0][0:2]		(5 pts)	lname.replace('cho','di')	

7
h02
CS8 M18

19. (5 pts) In Python, how do we define a string that consists of more than one line of text?

20. (5 pts) For the Python code in the left box, write the output in the right box

<pre>the_list = ["8", "16", "24", "32"] for course in the_list: print(course, end=" ")</pre>	
--	--

8
h02
CS8 M18

21. When you import the module `time`, the function `time.time()` gives “seconds since the epoch”, while `time.gmtime` gives a representation of the time in UTC.

a. (5 pts) What is “the epoch”?

b. (5 pts) What is UTC?

c. (5 pts) If you want the local time instead of UTC, what function of the `time` module do you call instead of `time.gmtime()`?

22. (5 pts) As shown on p. 106, if you enter `import time` at the `>>>` prompt, then enter the following:

```
>>> time.strftime('%A %b/%d/%y %I:%M %p', time.localtime())
```

the resulting output is the current date and time in the format:

```
Friday Aug/14/18 11:59 AM
```

Using the table of values on p. 107 as a reference, how would you change the call to `strftime` above to get the date in this format:

```
August 14, 2018
```