

SI 506 FALL 2017, MIDTERM EXAM

Name: SOLUTIONS

UMID: _____
(3 points) Your UMID is the 8-digit number on your M-Card.

Please write it at the top of every page of the exam.

You will earn 15 points by writing it on all 13 pages (including this one). These points cannot get you more than 100%, but if you do not earn full points on the exam otherwise, these 15 points will add to your score.

If you do not know your UMID, write your unique name (what comes before your @umich.edu e-mail address) on each page

This exam is worth 3,000 points. There are 23 questions.

We suggest that you take a quick pass to look through the entire exam and answer questions you are confident about before coming back to questions that will take you more time. There are questions near the end of the exam that are worth more points than other questions earlier in the exam; make sure you save time to address those.

This exam is closed book. You are allowed only one page of notes (double-sided). If a question seems unclear - please briefly write down any assumptions you feel are needed. If you think that there is a just-plain mistake/typo, check with an instructor.

Anywhere we ask you what will be printed out as a result of code being executed, if you think an error will be generated, you may write "error". You do not need to write out what the whole error message would be. **Note** that when string values are printed out on their own, quotation marks are not printed. You only see quotation marks as a result of printing when a string is *inside* a list or a dictionary or a tuple and *that* list/dictionary/tuple gets printed out.

CIRCLE your answers to multiple choice questions and make sure your answers are within the answer boxes given. We will not grade answers that are far outside the box because we scan the exams. Any text on a back of a page will NOT be graded.

By writing your name on this exam and turning it in, you are also asserting that all of the work you are turning in on this exam is your own, and you did not share answers with any other students or copy answers from someone else while taking the exam.

1. What is the type of mylst[1] after the following code executes? (50 points)

```
mylst = ['1', '2', '3']
```

- ☒ a. String
- b. Boolean
- c. List
- d. Dictionary
- e. Tuple
- f. Float
- g. Function
- h. None
- i. Integer

2. What is the type of mylst[1:] after the following code executes? (50 points)

```
mylst = ['1', '2', '3']
```

- a. String
- b. Boolean
- ☒ c. List
- d. Dictionary
- e. Tuple
- f. Float
- g. Function
- h. None
- i. Integer

3. What is the type of mylst[1]+mylst[2] after the following code executes? (50 points)

```
mylst = ['1', '2', '3']
```

- ☒ a. String
- b. Boolean
- c. List
- d. Dictionary
- e. Tuple
- f. Float
- g. Function
- h. None
- i. Integer

4. What is the type of **s.split** after the following code executes? (50 points)

→ function thing!

Remember: *split* is a method that can be invoked on strings. It can take no input OR can take input of a string that should be treated as a delimiter on which to split. **If it is invoked correctly**, the return value of invoking it is a list type.

```
s = '1 2 3 4 5'
```

- a. String
- b. Boolean
- c. List
- d. Dictionary
- e. Tuple
- f. Float
- ☒ g. Function
- h. None
- i. Integer

5. What is the type of **s.split('2')** after the following code executes? (50 points)

Remember: *split* is a method that can be invoked on strings. It can take no input OR can take input of a string that should be treated as a delimiter on which to split. **If it is invoked correctly**, the return value of invoking it is a list type.

```
s = '1 2 3 4 5'
```

- a. String
- b. Boolean
- ☒ c. List
- d. Dictionary
- e. Tuple
- f. Float
- g. Function
- h. None
- i. Integer

6. What is the output of the following expression? (50 points)

```
print(1 == 1)
```

- a. <class 'int'>
- ☒ b. True
- c. False
- d. 1
- e. 1.0
- f. No output; there is an error

7. What will print when the following code runs? (75 points)

```
d = {'a': 1, 'b': 'str', 'c': 0}
print(d[0])
```

- a. 'a'
- b. 'b'
- c. 'c'
- d. 'str'
- e. 0
- f. 1
- ☒ g. No output; there is an error

8. What will print when the following code runs? (75 points)

```
d = {'a': 1, 'b': 'str', 'c': 0}
print(d['b'])
```

- ☒ a. str
- b. 0
- c. 1 + 0
- d. <class 'function'>
- e. True
- f. False
- g. 1
- h. b
- i. No output; there is an error

9. The following code should print 4 but it currently produces the error: `TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'`. In **one sentence**, describe how would you fix it to provide the correct output. (75 points)

```
def add(x,y):
    sum = x + y

print(1 + add(1, 2)) # we want this to print: 4
```

Add return stmt to function
 → return sum
 (or equiv. effect)

answer

10. Which characters are printed when this code runs? **Select all that apply.** (100 points)

```
x = 10

if x <= 10:
    print('A')
if x >= 10:
    print('B')
    if x > 15:
        print('C')
    else:
        print('D')
    print('E')
elif x < 30:
    print('F')
else:
    print('G')
```

- ☒ a. A
- ☒ b. B
- ☐ c. C
- ☒ d. D
- ☒ e. E
- ☐ f. F
- ☐ g. G
- ☐ h. No output; there is an error

11. What will print when the following code runs? If there is an error, say so. (75 points)

```
x = [1, 3, 5, 2, 4]
y = x[1]
z = x[y]

print(z)
```

2

answer

12. What will print when the following code runs? If there is an error, write 'error'. (200 points)

```
umichDict = {
    "name": "University of Michigan",
    "founded": 1817,
    "type": "Public",
    "enrollment": 28395 + 15230, # undergrad + grad
    "nickname": "Wolverines",
    "colors": ["maize", "blue"]
}

print(len(list(umichDict.keys())))
print(umichDict["founded"])
```

6
1817

answer

13. What is the value of *i* after the following code runs? (50 points)

```
i = 2
while i < 4:
    i += 1
    print(i)
```

- a. 2
- b. 3
- ☒ c. 4
- d. 5
- e. Nothing; this is an infinite loop

14. Select every code snippet below that results in an infinite loop (meaning that the while loop never stops running once it begins). (200 points)


<pre>a = 1 while a != 10: print('A') a = a + 2</pre>	<pre>d = 0 while d > -10: print('D') d = d-1</pre>
<pre>b = 12 while True: print('B') b += 2 if b > 20: break</pre>	<pre>e = 14 while e < 40: e = 1 print 'E' e = e+1</pre>
<pre>c = 13 while c < 30: print('C')</pre>	<pre>f = 15 while f < 5: print('F')</pre>

as announced
assume
print
stmts.
are
correct for
Pyth3

← e.g.

15. What is the output of the following code? (200 points)

```
diction = {'b':'c', 'cd':'tape', 'hello': '12365'}
for k in list(diction.keys()):
    if 'b' in k:
        print(diction[k])
    elif 'z' in k:
        print(diction[k])
    else:
        print('No b and no z')
```



c

No b and no z

No b and no z

answer

16. What is the output of the following code? If there is an error, describe why. (100 points)

```
a = [[1,2,3], ['b','c']]
for x in a:
    if len(x) > 1:
        print(x[0])
```

1
b

answer

Suppose that a file called "test.txt" has the following contents:

```
This file has three lines.
Line two has five words.
This is the third line.
```

17. What will the following code print out? (100 points)

```
fname = 'test.txt'
print('fname')
```

fname

answer

18. What will the following code print out? (100 points)

```
fname = 'test.txt'
print(fname[2:])
```

st.txt

answer

19. What will the following code print out? (100 points)

```
fname = 'test.txt'
f = open(fname, 'r')
lines = f.readlines()
print(lines[1])
aline = lines[2]
print(aline[3])
```

Line two has five words.
5

answer

20. Define a function named `xplace` that accepts a string, and returns a string identical to the input string, **except** with the **first letter of the string (ONLY)** replaced by an 'x' (lowercase). Example calls are shown below.

HINT: String slicing and string concatenation may be useful here... (300 points)

```
def xplace( s ):
```

```
    return "x" + s[1:]
```

```
print xplace('hello heisman') # prints 'xello heisman'  
print xplace('MEME')         # prints 'xEME'
```

other possible sol'ns -

any parameter
body:

```
hs = "x" + s[1:] ...
```

etc.

all same idea, though

note:

• `replace()` method won't work

into
special
amendment

21. Define a function called **maybe_key** that should accept 1 required parameter, a string, and one optional parameter of a dictionary, whose default value is `{"ABCDE": "hello"}`. If the input string *is* a key in the input dictionary, the function should return the value associated with that key in the dictionary. However, if the input string is *not* a key in the dictionary, the function should return the boolean value **False**. There are example invocations below you should check out. (350 points)

```
def maybe_key(s, d = {"ABCDE": "hello"}):
    if s in d:
        return d[s]
    else:
        return False
```

answer

```
d = {"ABCDE": "ABC", 3, "ABCDEFGH", "hello": 22}
print(maybe_key("ABCDE")) # should print "hello"
print(maybe_key("ABCDE", d)) # should print "ABC"
print(maybe_key("hi there")) # should print False
print(maybe_key("a")) # should print False
print(maybe_key("yo", {"abc": 23, "yo": 32})) # should print
32
```

22. Write code that repeatedly asks the user to input numbers, and prints each number they enter, until the user enters the input **stop**. **stop** should also be printed, but after that, the loop should end. (250 points)

For example, suppose the user plans to enter the following values, in order: 4, 5, 2, "stop", 72. After entering "stop", the user is not asked for another number. The printed result should be:

4
5
2
stop

HINT: a while loop will be useful here...

```
inp = input("Pls enter number:")  
while inp != "stop":  
    print(inp)  
    inp = input("Pls enter number:")
```

→ ~~Check~~ the only correct way / answer!

answer

23. Given the following code, write additional code to determine and **print out the element that appears *least often* in the list `lst`.**

Your code must work for *any* list saved in the variable `lst`, not only this particular set of values – but you may assume that there will always be only one element that appears the fewest number of times in the list `lst`.

(For example, you can assume you will *not* have a list to deal with like `["hi", "bye"]` where both elements occur the fewest number of times.)

You do not have to define a function for this question, but you *may* define a function – if you do, make sure you write code that will print out the value you want to print out. (350 points)

```
lst = ["hi", "hey", "bye", "hi", "bye", "bye", "hi", "hi"]
```

```
counts = {}
for elem in lst:
    if elem in counts:
        counts[elem] = counts[elem] + 1
    else:
        counts[elem] = 1

c_keys = list(counts.keys())
least_so_far = c_keys[0]
for k in c_keys[1:]:
    if counts[k] < counts[least_so_far]:
        least_so_far = k

print(least_so_far)
```

answer

*not the only correct way.

