

CS303E Exam 3: Fall, 2025 § Version: A
Dr. Bill Young § Friday, December 5, 2025

Number

Name: _____ EID: _____

Read the questions carefully, and answer each question in the space provided. The last two pages are scratch paper; if you write anything there that needs to be graded, make that very clear. Don't write in the spaces marked "Page Total" or "Grades" at the bottom of this page.

Questions 1–5 (2 points each) require you to trace the behavior of some Python code and identify the output of that code. For each question, write the output for the code segment in the box. If executing the code gives an error or doesn't terminate, say so.

1.

```
def f(a, b=3, c=4):  
    return a + b + c
```

```
print( f(7, 2) )
```

--

2.

```
def countDown(n):  
    if n > 0:  
        print(n, end=" ")  
        countDown(n-1)
```

```
countDown(5)
```

--

3.

```
s = {1, 2, 3} | {3, 4, 5}  
print(s)
```

--

Grades: CS303E Exam 3

p1-6	p2-4	p4-20	p5-10	p6-10	Tot-50

```
4. myTuple = (10, 20, 30)
   myTuple[1] = 25
   print(myTuple)
```

```
5. s = "add"
   d = {'a':2, 'c':4}
   for ch in s:
       d[ch] = ( d[ch] + 1 if ch in d else 1 )
   print(d)
```

Questions 6–15 are multiple choice. Each counts 2 points. Each problem has a single answer, but be sure to read all choices. **Write the letter of the BEST answer in the box on page 4 in UPPERCASE; only answers written there will be counted.**

6. How many elements could be in the object returned by the following function:

```
def f(a, b):
    return {a + b, a - b, a * b}
```

A. 1 B. 2 C. 3 D. 2 or 3 E. 1, 2 or 3 F. it's an error

7. In a `print()` statement, what are the default values for the `sep` and `end` parameters, respectively?

A. "\n", " " B. " ", "\n" C. ",", "\t" D. " ", "\t"
E. they don't have defaults

8. If `L` is a list, what is a difference between: (i) `for x in L:`, and (ii) `for i in range(len(L)):`?

A. there is no difference
B. in the loop body you'd refer to elements of `L` as `x` in (i) and `L[i]` in (ii)
C. version (i) isn't legal
D. version (ii) isn't legal

9. This symbol marks the beginning of a comment in Python:

A. # B. * C. ** D. & E. \\\

Only answers copied to page 4 will be graded.

10. If `x` is a float, what does `round(x)` round to?
- A. up to the nearest integer.
 - B. down to the nearest integer.
 - C. to the nearest integer, but rounds up if the decimal part is ≥ 0.5 .
 - D. to the nearest integer, or to the nearest even integer if the decimal part equals 0.5.
 - E. to the nearest integer, but rounds down if the decimal part is ≤ 0.5 .
11. If `s1` and `s2` are sets of integers such that `s1.intersection(s2) == {1}`, which of the following must also be True?
- A. `s1.symmetric_difference(s2) != set()`
 - B. `s1.difference(s2) != set()`
 - C. `s2.difference(s1) != set()`
 - D. `s1.union(s2) != set()`
12. Assume `n` is a non-negative integer. What does the following recursive function return?

```
def mystery(n):  
    if n < 10:  
        return [n]  
    else:  
        return [n % 10] + mystery(n // 10)
```

- A. the sum of the digits of `n`
 - B. a list of the digits of `n`
 - C. a list of the digits of `n`, reversed
 - D. the maximum digit in `n`
13. Assume `n` is a non-negative integer. To return the sum of the squares of non-negative integers up to `n` what might be the missing expression?

```
def sumSquares(n):  
    if n == 0:  
        return 0  
    else:  
        return **missing expression**
```

- A. `n**2 + sumSquares(n-1)`
- B. `n**2 + sumSquares(n+1)`
- C. `n**2 + sumSquares(n)`
- D. `[n**2] + sumSquares(n-1)`
- E. `sumSquares(n**2 + n-1)`

Only answers copied to page 4 will be graded.

14. What is the syntax for adding a key/value pair to a dictionary?
- A. `dict.add(key, value)`
 - B. `dict.append(key, value)`
 - C. `dict[key] = value`
 - D. `dict.insert(key, value)`
15. What is the primary purpose of the `__init__` method in a Python class?
- A. to initialize the class attributes
 - B. to handle exceptions
 - C. to define the class methods
 - D. to justify the `self` parameter
 - E. to perform mathematical operations

Did you copy your answers to the boxes below? If not, they won't be graded!

6	7	8	9	10	11	12	13	14	15

16. (**Programming:** 10 points) On the next page complete the **recursive** function `transformString` that, given a string as input, returns a string like the input except:
- A. lowercase letters are replaced by their uppercase equivalent;
 - B. uppercase letters are replaced by their lowercase equivalent;
 - C. all other characters are left unchanged.

Non-recursive solutions will receive no credit. Don't use loops or a recursive helper function. This shows some sample behavior:

```
>>> transformString("abCDef123!@#$$&GHij")
'ABcdEF123!@#$$&ghIJ'
>>> transformString("")
''
>>> transformString("123#")
'123#'
```

```
def transformString( s ):
```

17. (**Programming:** 10 points) On the next page is the skeleton of a class, `ShoppingLine`, that models the checkout line at a store. In the line are customers with some number of items in their carts. Implement this as a private data member that is a list of customers with each customer represented as a pair (tuple) of (name, count) where count is a positive integer number of items. Here's some sample behavior:

```
>>> line = ShoppingLine([("Jim", 5), ("Andrea", 12), ("Rachel", 2)] )
>>> print(line)
Line: [('Jim', 5), ('Andrea', 12), ('Rachel', 2)]
>>> len(line)          # How many customers in the line?
3
>>> line.checkout()    # remove first customer, return their name
'Jim'
>>> print(line)
Line: [('Andrea', 12), ('Rachel', 2)]
>>> line.join("Jake", 7) # add a new customer to end of line
>>> print(line)
Line: [('Andrea', 12), ('Rachel', 2), ('Jake', 7)]
>>>
```

```
class ShoppingLine:
    def __init__( self, initialLine ):
        """Create a new ShoppingLine object with customers in initialLine."""

    def __len__( self ):
        """Return the number of customers in the line."""

    def join( self, name, itemCount ):
        """Add a new customer at the end of the line."""

    def checkout( self ):
        """If the line is empty, return None and leave the line unchanged.
        Otherwise, remove the first customer and return their name."""

    def __str__( self ):
        """Return a string for printing the line. Note str(L) will give
        you the string representation of a list L."""
```

Scratch paper: *if you use this space for anything that needs grading, please make that clear, e.g. “Answer continues on p. 7.”* If you didn’t use it at all or didn’t use it to record any of your answers, please carefully tear off this sheet before turning in your test and recycle it.

Scratch paper