

CS303E Week 6 Worksheet: Functions

Name: _____ EID: _____

Read the questions carefully, and answer each question in the space provided. Use scratch paper to do your work and then copy your answers neatly and legibly onto the test paper. Only answers recorded on the test paper will be graded.

1. (10 points: 1 point each) The following are true/false questions. **Write either T or F in the boxes at the bottom of page 1.** If there's any counterexample, it's false.
- (a) A function must always have an explicit return statement.
 - (b) A variable defined within a function is limited to that function and is not accessible outside of it.
 - (c) When a function lacks a return statement or includes a return statement without specifying a value, it implicitly returns 0.
 - (d) A function can have multiple parameters with the same name as long as they have different default values.
 - (e) Keyword arguments are passed to a function based on their order in the function's parameters, while positional arguments are explicitly specified by name in the function call.
 - (f) When you pass a mutable object as an argument to a function in Python, the function can modify the object.
 - (g) A **break** statement and a **return** statement can be used interchangeably.
 - (h) Functions can have multiple **return** statements, but at most one of them will be executed during the function's execution.
 - (i) Functions can return multiple values.
 - (j) A Python program does not have to contain a **main()** function.

a	b	c	d	e	f	g	h	i	j

Questions 2-7 are multiple choice. Each counts 2 points. **Write the letter of the BEST answer in the box on page 3. Please write your answer in UPPERCASE. Each problem has a single answer.**

2. What is the primary difference between using `return` and `print()`?
 - A. Both `return` and `print()` serve the same purpose, so you can use either one interchangeably.
 - B. `return` is used to exit a function and possibly pass a value to the caller, while `print()` is used to display output on the console.
 - C. `print()` is used to return values to the caller, while `return` is used to display output on the console.
 - D. `return` and `print()` are used for the same purpose, but `return` is used when working with numerical values, and `print()` is used for strings.
3. Which of the following is not an advantage of using a function?
 - A. Reusability of code.
 - B. Easier debugging and maintenance.
 - C. Increased program execution speed.
 - D. Improved code organization and readability.
4. What is a requirement for a function?
 - A. It must have at least one parameter.
 - B. It must contain at least one return statement.
 - C. It must be defined using the `def` keyword.
 - D. It must have default values for its parameters.
5. What is the significance of indentation in the definition of a function?
 - A. Indentation within a function is optional and doesn't affect the function's behavior.
 - B. Indentation is used to group code together and define the body of the function.
 - C. Indentation is necessary to declare function parameters.
6. Which of the following commands always cause you to exit a function?
 - A. `break`
 - B. `continue`
 - C. `return`
 - D. `go away`
 - E. both A and C

7. Which of the following, if any, is NOT true of Python functions?
- A. In a function call, you can mix keyword and positional arguments.
 - B. Every function must have an explicit return statement.
 - C. A Python program does not have to contain a `main()` function.
 - D. Defining a function and calling a function are the same thing.
 - E. All of the above are true.
 - F. Both B and D are false.

2	3	4	5	6	7

The following questions (2 points each) ask 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 provided. Don't worry about line breaks. If there's an error, say so, but you don't need to specify the type of error.

```
8. def lulu(buttercup, coco, jasper):  
    return jasper + buttercup / coco  
  
peanut = lulu(coco = 3, jasper = 2, buttercup = 6)  
print(peanut)
```

```
9. def coraline(wybie = 7):  
    wybie *= 2  
    return wybie * 2  
  
mister = round(6.5)  
taterTot = coraline(mister)  
pumpkin = coraline(coraline())  
print(taterTot, pumpkin)
```

```
10. def kookie(grey):  
    kitkat = True  
    basil = False  
    while not basil and kitkat:  
        break  
    return grey + 5  
  
print(kookie(10))
```

```
11. def josh(chica, apollo, coco, peaches):  
    return chica + apollo * coco * peaches
```

```
print(josh(chica = 5, peaches = 2, apollo = 2, 3))
```

```
12. def elmo():  
    cookieMonster = "chunky"  
    BIGbird = "squawky"  
    return BIGbird  
    return cookieMonster
```

```
myFavoriteStreet = elmo()  
print(myFavoriteStreet)
```

```
13. def marriage():  
    return "Lois", "Clark"
```

```
spouse1, spouse2 = marriage()  
print(spouse2, "<3", spouse1)
```

```
14. def countDracula():  
    count = 0  
    for num in range(10):  
        count += 1
```

```
print(count)
```