

Name: _____

Email Address (UW Net ID): _____@uw.edu

Section: _____

CSE 160 Summer 2023 - Midterm Exam

Instructions:

- You have **60 minutes** to complete this exam.
- The exam is **closed book**, including no calculators, computers, phones, watches or other electronics.
- You are allowed a single sheet of notes for yourself.
- We also provide a syntax reference sheet.
- Turn in *all sheets* of this exam, together and in the same order when you are finished.
- When time has been called, you must put down your pencil and stop writing.
 - **Points will be deducted if you are still writing after time has been called.**
- You may only use parts and features of Python that have been covered in class.
- All questions assume Python version 3.7, as we have been using all quarter.
- You may ask questions by raising your hand, and a TA will come over to you.

Good luck!

| Question | Points |
|--------------|-----------|
| Question 1 | 7 |
| Question 2 | 6 |
| Question 3 | 4 |
| Question 4 | 5 |
| Question 5 | 5 |
| TOTAL | 28 |

Question 1 [7 Pts]:

For each of the expressions below, write what it evaluates to and the type of that value (float, int, string, or bool). You should specify any floating point values only to one decimal point (e.g., 0.5). Remember to add quotation marks if the expression evaluates to a string.

If an error is produced, write "Error." under the evaluation column and leave the type column blank

| Expression | Evaluation | Type |
|---------------------------------------|------------|------|
| "Great! " + str(100) + " %" | | |
| (2.5 * 2) + 1 | | |
| l = [1, 2, 3, 4] a = 4 a + l[a] | | |
| "awesome"[2] + "gg" | | |

Question 1b :

Given the following dictionary, what will the following expressions evaluate to?

```
produce = {  
    "oranges": 5.99,  
    "olives": 12.5,  
    "tomatoes": 4.12,  
    "peaches": 2.2  
}
```

| Expression | Result |
|-----------------------------|--------|
| produce["olives"] * 2 | |
| "peaches" in produce.keys() | |
| produce["kale"] | |

Question 2 [6 Pts]:

Given a list of animals `animals` and a list of characters `letters`, find and print the number of animals in `animals` that have at least one of the letters in `letters`. For example, given:

```
animals = ["hare", "mouse", "eagle", "sparrow", "horse"]
letters = ["a", "l", "h"]
```

Your code should print `4`.

This is because "hare" has an "a" and "h", "eagle" has an "a", "sparrow" has an "a", and "horse" has a "h". "mouse" has none of the letters in `letters`. Note that even though "hare" has 2 of the letters in `letters`, it only contributes 1 to the overall count.

Hint: Think about ways of keeping track of whether or not any relevant letters exist in each animal's name.

Write your code in the space below.

```
animals = ["hare", "mouse", "eagle", "sparrow", "horse"]
letters = ["a", "l", "h"]
count = 0
```

Question 3 [4 Pts]:

Given the following program, what will be printed out when the program is run?

```
def letter_code(num1, num2):  
  
    result = "A"  
    if num1 * num2 < 12:  
        result = "B"  
  
    if num1 % 2 == 0:  
        result = "C"  
    elif num1 // num2 == 3:  
        result = "D"  
  
    return result  
  
result1 = letter_code(2, 4)  
result2 = letter_code(3, 7)  
result3 = letter_code(9, 3)  
result4 = letter_code(5, 2)  
  
print(result1)  
print(result2)  
print(result3)  
print(result4)
```

Question 4 [5 Pts]:

An underwater drone has recently been developed to explore the bermuda triangle. It can only go to a maximum depth of 500m (inclusive) and it has to stay at or below 15 knots of speed due to wildlife. If the submarine violates these specifications, a warning will be generated

Given two lists of integers, `depth` and `knots`, where each index represents a trip, write a function `assess_trip` that returns a list with a statement corresponding to each trip.

Arguments:

`depth`: a list of integers to specifying maximum depth of each trip

`knots`: a list of integers to specifying speed of each trip

Returns:

A list of strings with the following value at each index:

“Okay” if the submarine is within the limits.

“Warning” if the submarine is outside of the limits

Example:

```
depth = [100, 500, 200, 600]
```

```
speed = [15, 10, 20, 65]
```

```
assess_trip(depth, speed) returns ["Okay", "Okay", "Warning",  
"Warning"]
```

Explanation:

- At index 0, the drone goes 100 meters and 15 knots, which is within the limits
- At index 1, the drone goes 500 meters and 10 knots, which is within the limits
- At index 2, the drone goes 200 meters and 20 knots, which exceeds the speed limit
- At index 3, the drone goes 600 meters and 65 knots, which exceeds the depth limit

```
def assess_trip(depth, speed):  
    # Write your code below this line
```

Question 5 [5 Pts]:

Before this code is executed, assume that the file "language.txt" file is empty. What would be the output of the code below?

```
file = open("language.txt", "w")
file.write("Python\n")
file.write("JavaScript\n")
file.write("Java\n")
file.write("C++\n")
file.write("Swift\n")
file.close()
```

```
file = open("language.txt", "r")
index = 0
for language in file:
    if index % 2 == 0:
        print(language)
    index = index + 1
file.close()
```

Extra Credit [1 Pt]:

Name two of your TAs (just the first name is fine)