Name: Nishant

Email: Nishant575051@gmail.com

Contact: 9654546020

Task 1

Question 1:

Review the following codes, find and fix errors and explain the error

Code Snippet 1: Variable Name Typo
Code:

```python
number_of_apples = 5
print(number_of_apple)
```

Error to Expect: NameError
Hint: Ensure that variable names are spelled consistently throughout your code. Did you reference the variable correctly when trying to print it?

Solution:

Explanation: due to word “s” not match with variable error has been occurred.

```
number_of_apples = 5
print(number_of_apples)
```

```
5
```

Question 2:

Code Snippet 2: Accessing List Elements Out of Range
Code:

```python
fruits = ["apple", "banana", "cherry"]
print(fruits[3])
```

Error to Expect: IndexError
Hint: List indices start at 0. If you're trying to access the third element, what index should you use? Also, consider how many elements are actually in the list.

Solution:

Explanation: as list index started from 0 to nth and total list fruits contain 3 values. Index of the fruits is : 0,1,2.

```
]: fruits = ["apple", "banana", "cherry"]
   print(fruits[2])
   print("total value in the list fruits: ",len(fruits))

   cherry
   total value in the list fruits:  3
```

Question 3:

Debugging Exercise 3: Function Not Behaving as Expected

```python
def find_average(numbers):
    sum = 0
    for number in numbers:
        sum += number
    average = sum / len(numbers)
    return average

numbers = [1, 2, 3, 4, 5, "6"]
average = find_average(numbers)
print(f"The average is: {average}")
```

Error to Expect: TypeError
Hint: This code attempts to calculate the average of a list of numbers. However, one of the elements in the list is not a number in the conventional sense. How does Python treat different data types when performing arithmetic operations? Consider how you can ensure all elements are appropriately handled for the calculation.

Solution:

Explanation: as list “number” contain string value “6”. While calculating average though function “find_average”. We need to convert all the value into integer.

```python
def find_average(numbers):
    sum = 0
    for num in numbers:
        sum += int(num)
    average = sum / len(numbers)
    return average

numbers = [1, 2, 3, 4, 5, "6"]
average = find_average(numbers)
print(f"The average is: {average}")
```

```
The average is: 3.5
```

Question 4:

Exercise 4: Incorrect Dictionary Usage
Code:

```python
def update_record(records, name, score):
    if name in records:
        records[name].append(score)
    else:
        records[name] = score

student_records = {"Alice": [88, 92], "Bob": [70, 85]}
update_record(student_records, "Charlie", 91)
update_record(student_records, "Alice", 95)

print(student_records)
```

Error to Expect: AttributeError
Hint: This function is supposed to update a dictionary containing student names as keys and a list of scores as their values. When adding a score for a student not previously in the records, the code does not correctly handle the assignment. How should you structure the value associated with each key in the dictionary, especially when introducing a new key?

Solution: as code is running successfully so there is no error. For reference see the below running code.

```
]: def update_record(records, name, score):
       if name in records:
           records[name].append(score)
       else:
           records[name] = score

   student_records = {"Alice": [88, 92], "Bob": [70, 85]}
   update_record(student_records, "Charlie", 91)
   update_record(student_records, "Alice", 95)
   print(student_records)

   {'Alice': [88, 92, 95], 'Bob': [70, 85], 'Charlie': 91}
```