TASK - 1

D.Sai Tejaswini

Code Snippet 1 : Variable Name Typo

Code :

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

Corrected Code : NameError

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

Explanation :

The variable '**number_of_apple**' was misspelled. It should be '**number_of_apples**' to match the variable defined earlier.

Code Snippet 2: Accessing List Elements out of Range

Code:

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

Corrected Code : IndexError

```
fruits = ["apple", "banana", "cherry"]
print(fruits[2])
```

Explanation :

List indices in Python start at 0. The third element in the list is at index 2, not 3. Accessing '**fruits[3]**' would be out of range, causing an '**IndexError**'.

Debugging Exercise 3 : Function Not Behaving as Expected

Code :

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

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

Corrected Code : TypeError

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

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

Explanation:

To find the average of a list of numbers, ensure all elements are numerical by converting any strings to integers or floats. Sum all the elements and then divide the total by the number of elements in the list, which can be obtained using the **'`len()`'** function. This ensures accurate arithmetic operations and prevents type errors.

Exercise 4: Incorrect Dictionary Usage

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)
```

Corrected Code : AttributeError

```
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)
```

Explanation :

When adding a new student, the code incorrectly assigned a single score instead of initializing it as a list. The corrected code initializes the value as a list containing the score.