**Code Snippet 1(Variable Name Typo):**

**Explanation of the error**:

The error in the code snippet is the name of the variable. The variable name “number\_of\_apples” is the actual name during the variable assignment, the number 5 is being assigned to the variable “number\_of\_apples”.

But in the print statement the name of the variable is wrong as it is written as: print(number\_of\_apple). But the actual variable name is: number\_of\_apples where in the print statement the letter ‘s’ is missing.

**Corrected Code:**

number\_of\_apples = 5

print(number\_of\_apples)

**Code Snippet 2(Accessing List Elements Out of Range):**

**Explanation of the error:**

Here in the code snippet, we are trying to access and display the 3rd element in a list, but here in the list we have 3 elements and they are “apple”, “banana” and “cherry”. To access we need to know that, the indexing always starts from 0. Therefore, if we want to access the last element then, it would be of index 2 and not 3.

**Corrected Code:**

fruits = [“apple”, “banana”,“cherry”]

print(fruits[2])

**Code Snippet 3(Function Not Behaving as Expected):**

**Explanation of the error:**

Here in the code snippet, we can see that we are using functions to add up the numbers from a list, and we also know that, we can work on the list with different data types. For example, we can declare integer, character, float, etc. at the same time inside a list. In the given code we can see that one of the elements in the list is not an integer rather it is a character where the interpreter cannot add a character with an integer that is why the code snippet is showing a TypeError. Here, is the original variable with the error:

numbers = [1,2,3,4,5,”6”], and the corrected format should be, numbers = [1,2,3,4,5,6]. No inverted commas must be given in order to run the program smoothly.

**Corrected Code:**

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}")

**Code Snippet 4(Incorrect Dictionary Usage):**

**Explanation of the error:**

In the code, most of the format is okay, but if we look closely we can see that the elements that are being shown in the output is not matching. If we run the original code the output shown is: {'Alice': [88, 92, 95], 'Bob': [70, 85], 'Charlie': 91} where the record of Charlie is different than the other two students. Here, we can see that the record is not in a list, but rather it is directly declared as an integer. To correct the error, we have to go to the line where assignment of the records is being done: “records[name] = score”. Here, in the code statement we can see that the variable score should be declared as a list not an integer. So, in order to correct it here is the corrected statement: “records[name] = [score]”. Now, the output will be: {'Alice': [88, 92, 95], 'Bob': [70, 85], 'Charlie': [91]}

**Corrected Code:**

def update\_record(records, name, score):

if name in records:

records[name].append(score)

else:

records[name] = [score] #here is the code statement that was wrong.

student\_records = {"Alice": [88,92], "Bob": [70,85]}

update\_record(student\_records, "Charlie", 91)

update\_record(student\_records, "Alice", 95)

print(student\_records)