

## Task-1

Code Snippet 1: Variable Name Typo

Code:

```
python

number_of_apples = 5
print(number_of_apple)
```

Error: name error

Corrected code:

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

the variable inside the print should be of the same name as in the declaration of the variable

.....

Code Snippet 2: Accessing List Elements Out of Range

Code:

```
python

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

Error : index out of bound error

Corrected code is :

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

Indexing starts from 0 and in the given snippet the maximum index is 2 representing "cherry".

.....

## 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 : Type error for the array elements.

---

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) ")
```

---

As all the elements are of type int and one element is of type character, the error is about whether to consider the array as an int or a character array.

.....

## 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 : Attribute error

Corrected 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)
```

For existing students, the score is appended to their list of scores and for the new student a new list is created containing the first score and assigned to the student's name in the directory.

.....