Vaultofcodes

Assignment 1 - Task 1

Code Snippet-1: Variable Name Type

Error code:

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

Type of Error:

The error in the above code is NameError, name number_of_apple is not define

Correct code:

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

Explanation of error:

It happens because the variable you are trying to print is named number_of_apples, but in the print statement, you used number_of_apple (missing the "s" at the end). To fix the error, you need to ensure that the variable name in the print statement matches the one defined earlier

Code Snippet-2: Accessing List Elements Out of Range

Error Code:

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

Type of Error:

The error in the above code is IndexError , list index out of range

Correct code:

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

Explanation of error:

Error occurs because the list fruits only has three elements, indexed as 0, 1, and 2. When you try to access fruits[3], you are attempting to access the fourth element, which does not exist in the list, causing the error. To fix this, you should use an index within the valid range, such as fruits[0], fruits[1], or fruits[2].

Debugging Exercise 3: Function Not Behaving as Expected

Error 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}")
```

Type of error:

The error in the above code is TypeError

Correct code:

```
def find_average(numbers):
    sum = 0
    for number in numbers:
        sum += int(number)  # Convert each number to integer
    average = sum / len(numbers)
    return average

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

Explanation of error:

The list numbers contains a string ("6"), causing a TypeError when attempting to add it to the running total sum.To fix the error, you should ensure all elements in the list are integers.

Exercise 4-Incorrect Dictionary Usage

Error 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)
```

Type of error:

The error in the above code is AttributeError

Correct code:

```
def update_record(records, name, score):
   if name in records:
      records[name].append(score)
   else:
      records[name] = [score]  # Assign a list containing the 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 of error:

When adding a new name to the records dictionary, the score is directly assigned rather than being placed in a list.This results in inconsistent data structure where some entries are lists and others are not. It will cause issues if you later try to treat all entries as lists.To fix the error, the score for new names should be assigned as a list containing the score.