ASSIGNMENT – 1

TASK- 1

1.
```
number_of_apples=5

print(number_of_apple)
```

Error : The error in the above code is a typo in the variable name used in the print statement.

Corrected code:

```
number_of_apples=5

print(number_of_apples)
```

Explaination of error: The error occurred because the variable name used in the print statement didn't match the actual variable name. Python is case-sensitive, so `number_of_apple' and `number_of_apples' are treated as different variables.

2.
```
fruits = ["apple", "banana", "cherry"]

print(fruits[3])
```

Error: The error in the above code is an "IndexError".

Corrected code:

```
fruits = ["apple", "banana", "cherry"]

print(fruits[2])
```

Explaination of error: The error is that the index you're trying to access, `fruits[3]`, is out of range. In Python, indexing starts from 0, so the last element in the list would be at index 2, not 3. The above code will print the value "cherry", which is the third element in the list. Remember, when accessing elements in a list, the index should be within the range of the list's length.

3.

```
 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: The error in the above code is an "TypeError".

Corrected code:

```
def find_average(numbers):
    total = 0
    count = 0
    for number in numbers:
        if isinstance(number, (int, float)):
            total += number
            count += 1
    if count == 0:
        return 0
    average = total / count
    return average
numbers = [1, 2, 3, 4, 5, "6"]
average = find_average(numbers)
print(f"The average is: {average}")
```

Explaination of error: This error occurs when you try to perform an operation on incompatible data types. In this case, the error is caused by trying to add a string ("6") to the sum of the numbers.Python treats different data types differently when performing arithmetic operations. When you try to add a string to a number, Python raises a `TypeError` because it doesn't know how to perform addition between a string and a number.To fix this error, we need to ensure that all elements in the `numbers` list are numbers (integers or floats) before performing the addition operation.

4.

```
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: The error in the above code is an "AttributeError".

Corrected code:

```
def update_record(records, name, score):
    if name in records:
        records[name].append(score)
    else:
        records[name] = [score]  # Ensure that the value is always a list
student_records = {"Alice": [88, 92], "Bob": [70, 85]}
update_record(student_records, "Charlie", 91)
update_record(student_records, "Alice", 95)
print(student_records)
```

Explaination of error: The AttributeError occurs because when a new student ("Charlie") is added to the student_records dictionary, the function assigns a single score directly to the student's name. Later, when attempting to append a score for an existing student ("Alice"), the code expects the value associated with each student's name to be a list, but for "Charlie", it's a single score. Thus, an AttributeError is raised when trying to use the append() method on a non-list object.