Assignment-1

VaultofCodes

Name- Suman Kumar Gochhayat

GitHub Link- https://github.com/ssgochhayat/Vaults_of_codes_Internship.git

Assignment 1: Code Review and Error Correction

Objective: Identify, correct, and explain errors in Python code snippets to reinforce understanding of syntax and logic.

Instructions:

For each code snippet below, do the following:

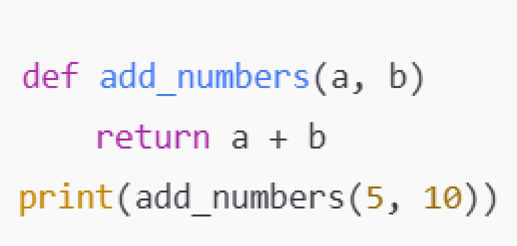
1. **Identify** the errors in the code.

2. **Rewrite** the corrected code.

3. **Explain** each error and why the correction works.

Code Snippets

1. **Code Snippet 1**



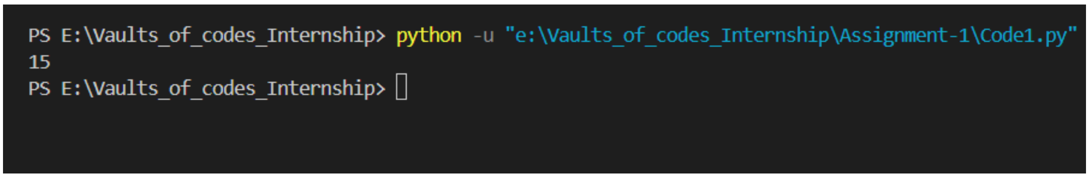
```
def add_numbers(a, b)
    return a + b
print(add_numbers(5, 10))
```

Error : Missing Indentation symbol colon (:) at the end of the def line.

Correct Code:

```
def add_numbers(a, b):
    return a + b
print(add_numbers(5, 10))
```

Output:

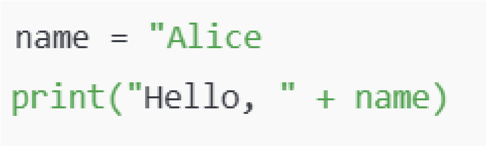


```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code1.py"
15
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

In Python, a colon (:) is required at the end of a function definition line to indicate the start of the function body. Without the colon, Python raises a SyntaxError because it expects a block to follow but cannot recognize where it begins.

2. **Code Snippet 2**



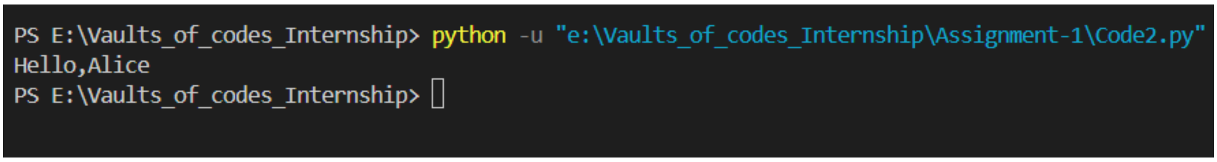
```
name = "Alice
print("Hello, " + name)
```

Error : Missing closing quotation mark (") at the end of the string in the print() function.

Correct Code:

```
name="Alice"
print("Hello,"+name)
```

Output:

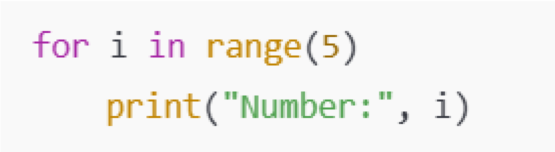


```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code2.py"
Hello,Alice
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

In Python, all string literals must be properly enclosed within matching quotation marks — either single (') or double ("), which causes a SyntaxError.

3. **Code Snippet 3**



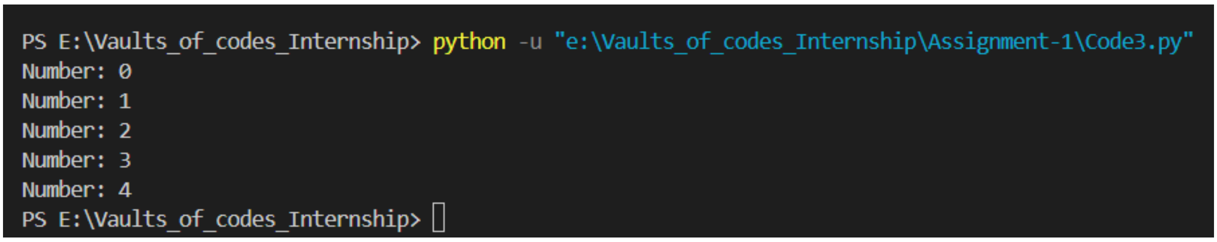
```
for i in range(5)
    print("Number:", i)
```

Error : The print() statement is not properly indented inside the for loop block.

Correct Code:

```
for i in range(5):
  print("Number:", i)
```

Output:



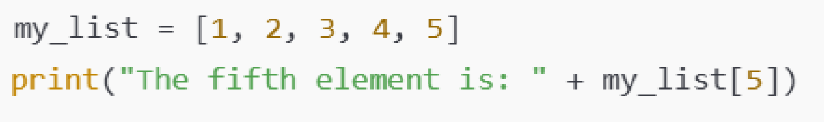
```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code3.py"
Number: 0
Number: 1
Number: 2
Number: 3
Number: 4
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

In Python, indentation is mandatory to define blocks of code. The print() statement must be indented under the for loop to show that it belongs to that loop. Without indentation, Python raises an IndentationError or executes it incorrectly.

Assignment-1

4. **Code Snippet 4**



```
my_list = [1, 2, 3, 4, 5]
print("The fifth element is: " + my_list[5])
```

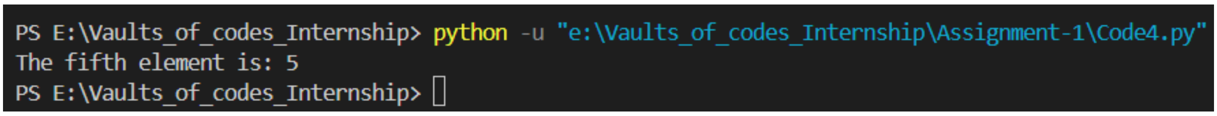
Error :

- IndexError: my_list[5] is out of range because list indices start at 0.
- TypeError: Trying to concatenate a string with an integer using +.

Correct Code:

```
my_list = [1, 2, 3, 4, 5]
print("The fifth element is: " + str(my_list[4]))
#or print("The fifth element is: " + str(my_list[(len(my_list)-1)]))
```

Output:

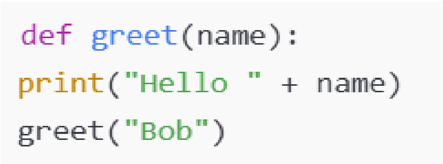


```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code4.py"
The fifth element is: 5
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

- Python lists use zero-based indexing, meaning the first element is at index 0. So the fifth element is at index 4, not 5. Accessing my_list[5] raises an IndexError.
- You cannot concatenate a string and an integer directly with +. Python raises a TypeError. You must first convert the integer to a string using str(), or use a comma , in the print() function .

5. **Code Snippet 5**



```
def greet(name):
print("Hello " + name)
greet("Bob")
```

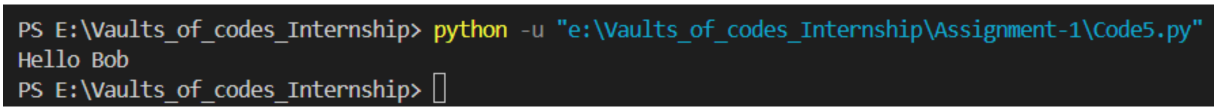
Error :

The print() statement inside the function is not indented. Python expects the function body to be indented or write pass.

Correct Code:

```
def greet(name):
    print("Hello " + name)
greet("Bob")
```

Output:

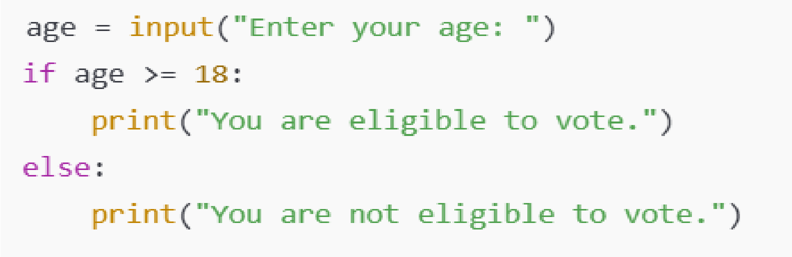


```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code5.py"
Hello Bob
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

In Python, the function body must be indented. Without indentation, Python raises an IndentationError. By indenting the print() line inside the greet() function, we properly define the code that should run when the function is called.

6. **Code Snippet 6**



```
age = input("Enter your age: ")
if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

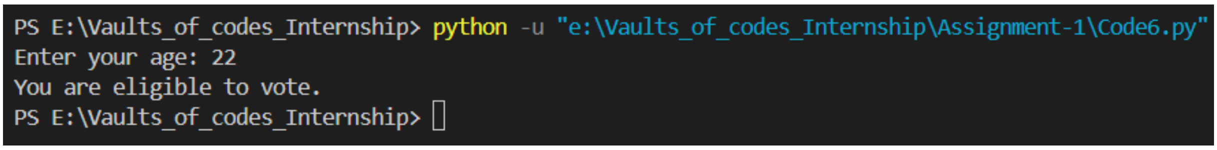
Error :

input() function returns a string, so comparing it directly with an integer (age >= 18) causes a TypeError.

Correct Code:

```
age = int(input("Enter your age: "))
if age >= 18:
   print("You are eligible to vote.")
else:
   print("You are not eligible to vote.")
```

Output:

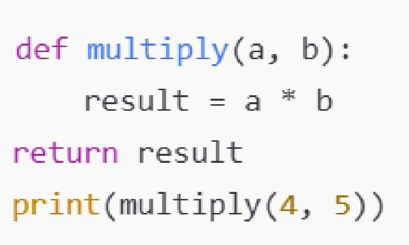


```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code6.py"
Enter your age: 22
You are eligible to vote.
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

input() always returns a string by default. Comparing a string with an integer (like age >= 18) will raise a TypeError. To fix this, we convert the input to an integer using int().

7. **Code Snippet 7**



```
def multiply(a, b):
    result = a * b
return result
print(multiply(4, 5))
```

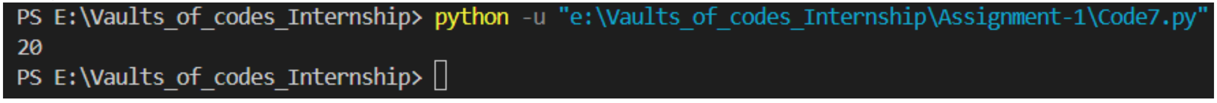
Error :

return statement is not indented, so it's outside the function block, which causes an IndentationError.

Correct Code:

```
def multiply(a, b):
   result = a * b
   return result
print(multiply(4, 5))
```

Output:

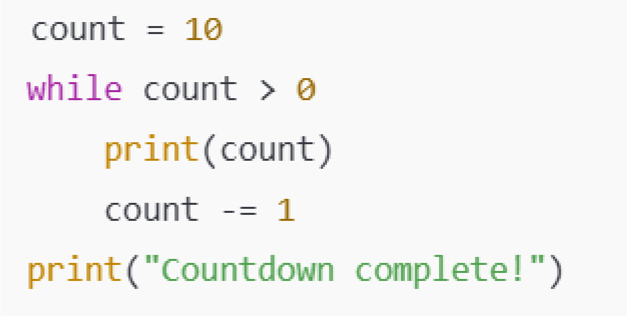


```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code7.py"
20
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

The return statement must be indented to show that it belongs to the function multiply. Without indentation, Python throws an IndentationError, because it expects the return statement to be inside the function but finds it outside.

8. **Code Snippet 8**



```
count = 10
while count > 0
    print(count)
    count -= 1
print("Countdown complete!")
```

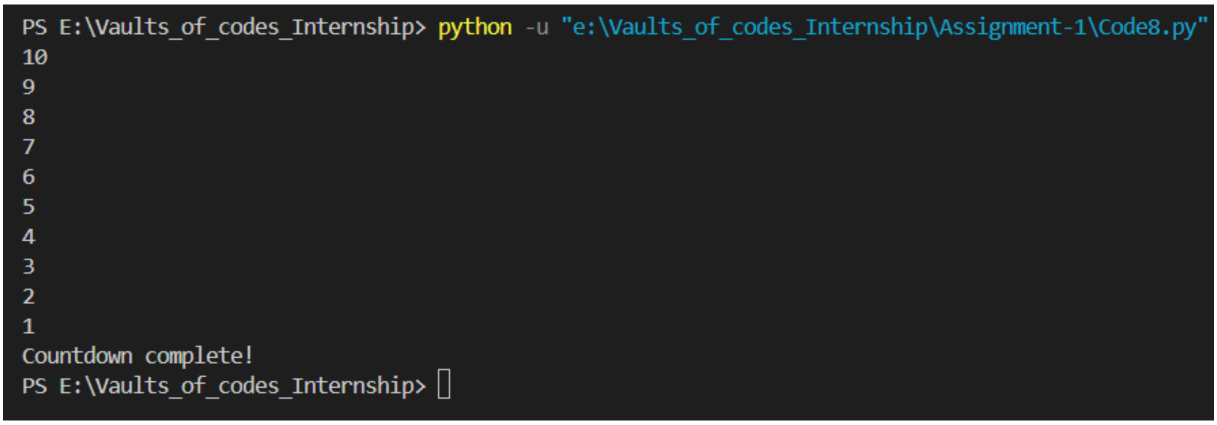
Error :

Missing Indentation symbol colon (:) at the end of the while loop condition, which causes an IndentationError.

Correct Code:

```
count = 10
while count > 0:
   print(count)
   count -= 1
print("Countdown complete!")
```

Output:



```
PS E:\Vaults_of_codes_Internship> python -u "e:\Vaults_of_codes_Internship\Assignment-1\Code8.py"
10
9
8
7
6
5
4
3
2
1
Countdown complete!
PS E:\Vaults_of_codes_Internship> 
```

Error Explanation:

In Python, a colon (:) is required at the end of while loop condition line to indicate the start of the while loop body. Without the colon, Python raises a Indentation Error because it expects a block to follow but cannot recognize where it begins.

--X--