Assignment – 7.1

Error Debugging with AI: Systematic approaches tofinding and fixing bugs

NAME: I.Harshith

HT.NO: 2303A51624

BATCH – 08

Lab Objectives:

- To identify and correct syntax, logic, and runtime errors in

Python programs using AI tools.

- To understand common programming bugs and AI-assisteddebugging suggestions.
- To evaluate how AI explains, detects, and fixes differenttypes of coding errors.
- To build confidence in using AI to perform structureddebugging practices.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Use AI tools to detect and correct syntax, logic, and runtimeerrors.
- Interpret AI-suggested bug fixes and explanations.
- Apply systematic debugging strategies supported by AI-generated insights.
- Refactor buggy code using responsible and reliableprogramming patterns.

Task - 01 (Syntax Errors Missing Parentheses in PrintStatement)

Buggy Code:

```
#Missing parentheses in print statement
def greet():
print "Hello, AI Debugging Lab!"
greet()
```

AI Prompt:

Why is this code giving an error? Fix the syntax and explain the issue.

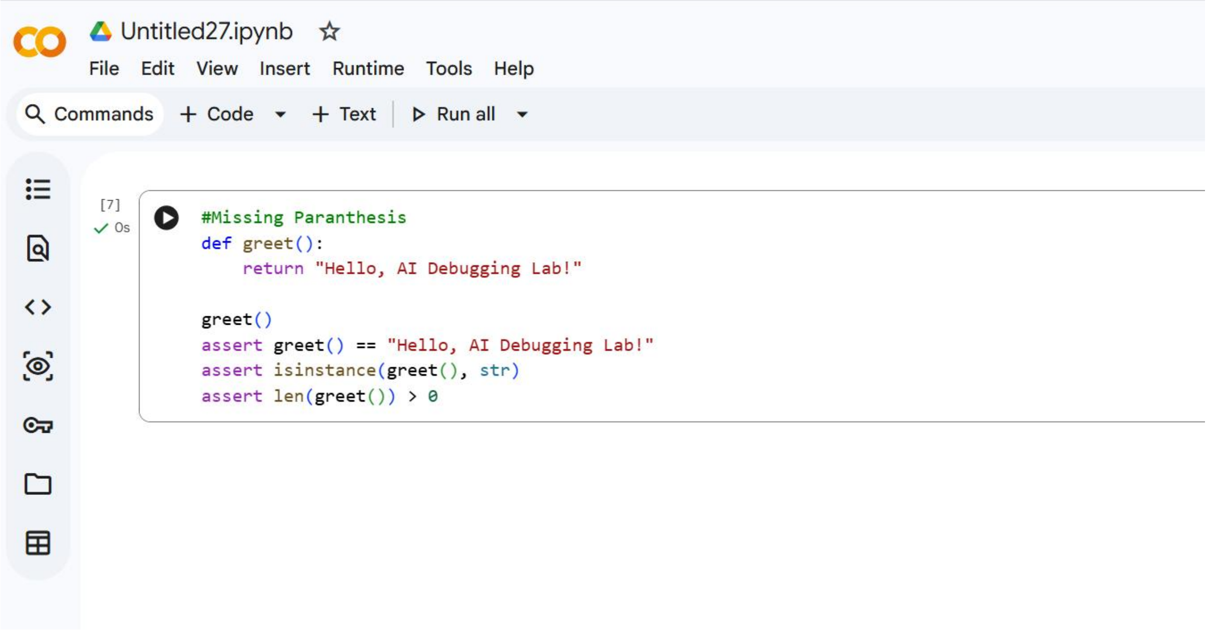
AI Explanation:

The code is giving a SyntaxError because you're using Python 2 syntax for the print statement. In Python 3, print is a function and requires parentheses around its arguments.

Corrected Code:

```
#Missing Paranthesis
def greet():
    return "Hello, AI Debugging Lab!"
greet()
```

Output:


Untitled27.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[7]
0s

```
#Missing Paranthesis
def greet():
    return "Hello, AI Debugging Lab!"

greet()
assert greet() == "Hello, AI Debugging Lab!"
assert isinstance(greet(), str)
assert len(greet()) > 0
```

Task - 02 (Incorrect condition in an If Statement)

Buggy Code:

```
def check_number(n):
if n = 10:
return "Ten"
else:
return "Not Ten"
```

AI Prompt:

Why does using = in the if condition cause an error? Fix it.

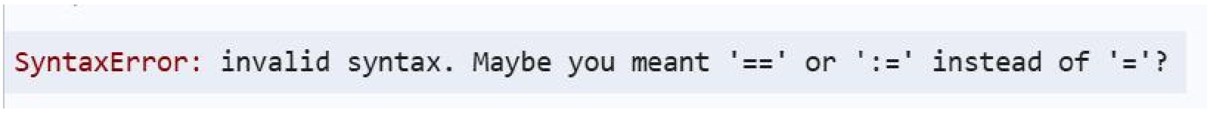
AI Explanation:

The error you're seeing, a SyntaxError, is because you're using a single equals sign = in your if statement condition: if n = 10:. In Python, a single equals sign is used for assignment, but for comparison, you need to use a double equals sign ==. I will fix this by changing = to ==.

Corrected Code:

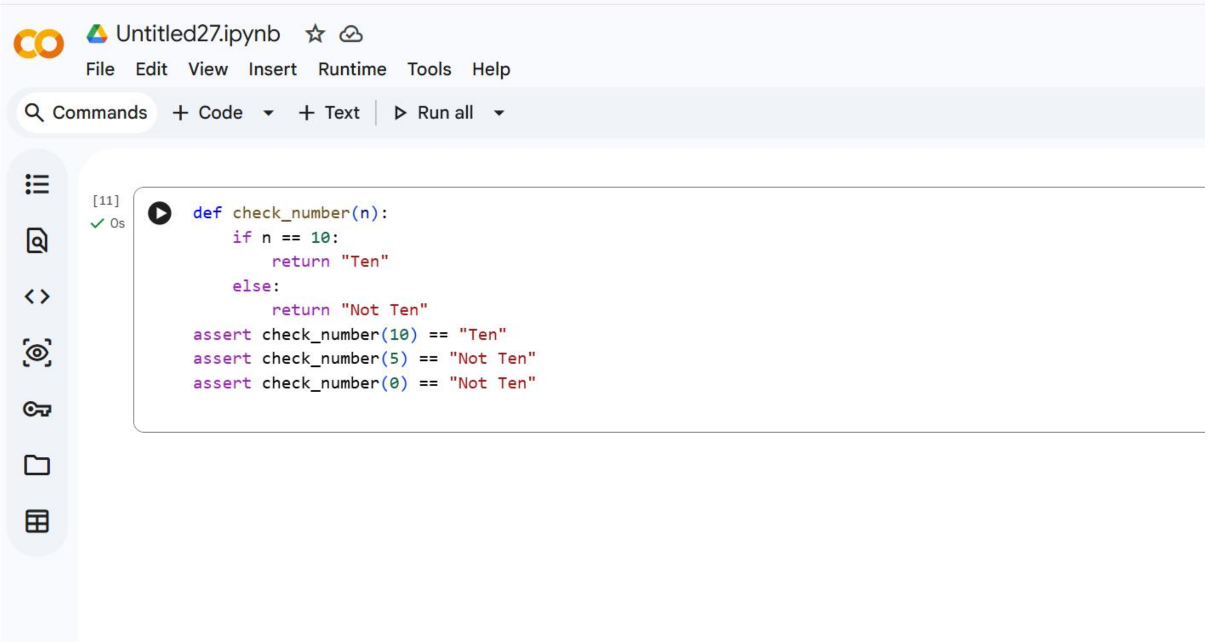
```
def check_number(n):
    if n == 10:
        return "Ten"
    else:
        return "Not Ten"
```

Error:



```
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

Output:



```
def check_number(n):
    if n == 10:
        return "Ten"
    else:
        return "Not Ten"
assert check_number(10) == "Ten"
assert check_number(5) == "Not Ten"
assert check_number(0) == "Not Ten"
```

Task - 03 (Runtime Error – File Not Found)

Buggy Code:

```
def read_file(filename):
with open(filename, 'r') as f:
return f.read()
print(read_file("nonexistent.txt"))
```

AI Prompt:

Add safe error handling using try-except and give user-friendly messages.

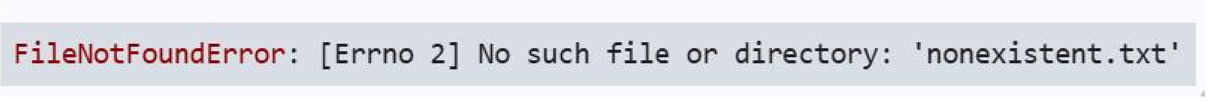
AI Explanation:

The error you're encountering is a FileNotFoundError. This means that your program is trying to open a file called nonexistent.txt, but it cannot find a file with that name in the directory where the script is running. I'll modify the code to handle this error gracefully using a try-except block, so it will print a message if the file is not found, instead of stopping execution.

Corrected Code:

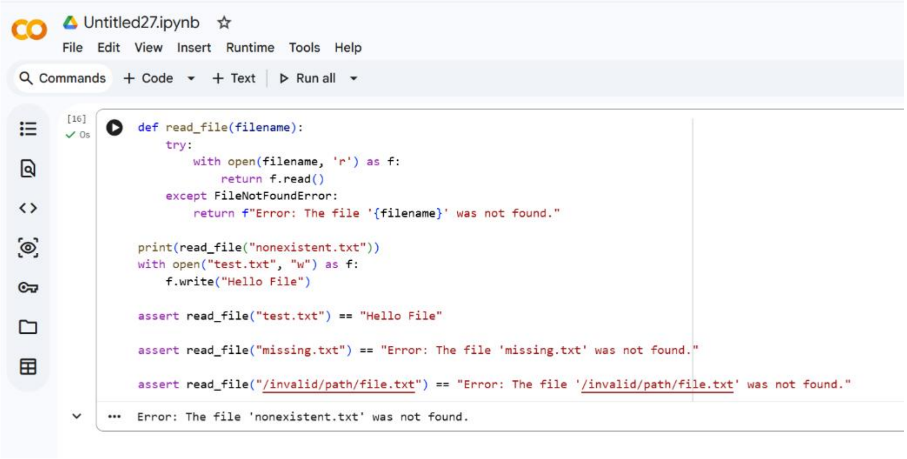
```
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return "Error: File not found."
    except Exception:
        return "Error: Invalid file path."
print(read_file("nonexistent.txt"))
```

Error:



```
FileNotFoundError: [Errno 2] No such file or directory: 'nonexistent.txt'
```

Output:



```
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return f"Error: The file '{filename}' was not found."

print(read_file("nonexistent.txt"))
with open("test.txt", "w") as f:
    f.write("Hello File")

assert read_file("test.txt") == "Hello File"

assert read_file("missing.txt") == "Error: The file 'missing.txt' was not found."

assert read_file("/invalid/path/file.txt") == "Error: The file '/invalid/path/file.txt' was not found."
```

```
Error: The file 'nonexistent.txt' was not found.
```

Task – 04(Calling a Non-Existent Method)

Buggy Code:

```
class Car:

def start(self):

return "Car started"

my_car = Car()

print(my_car.drive()) # drive() is not defined
```

AI Prompt:

Fix the AttributeError and explain the issue.

AI Explanation:

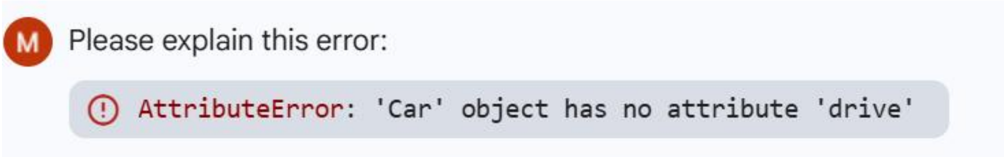
You're getting an AttributeError because you're trying to call a method named drive() on your Car object, but the Car class doesn't have a drive() method defined. It only has a start() method. I will fix this by changing my_car.drive() to my_car.start().

Corrected Code:

```
class Car:
    def start(self):
        return "Car started"

    def drive(self):
        return "Car is driving"
my_car = Car()
print(my_car.drive())
```

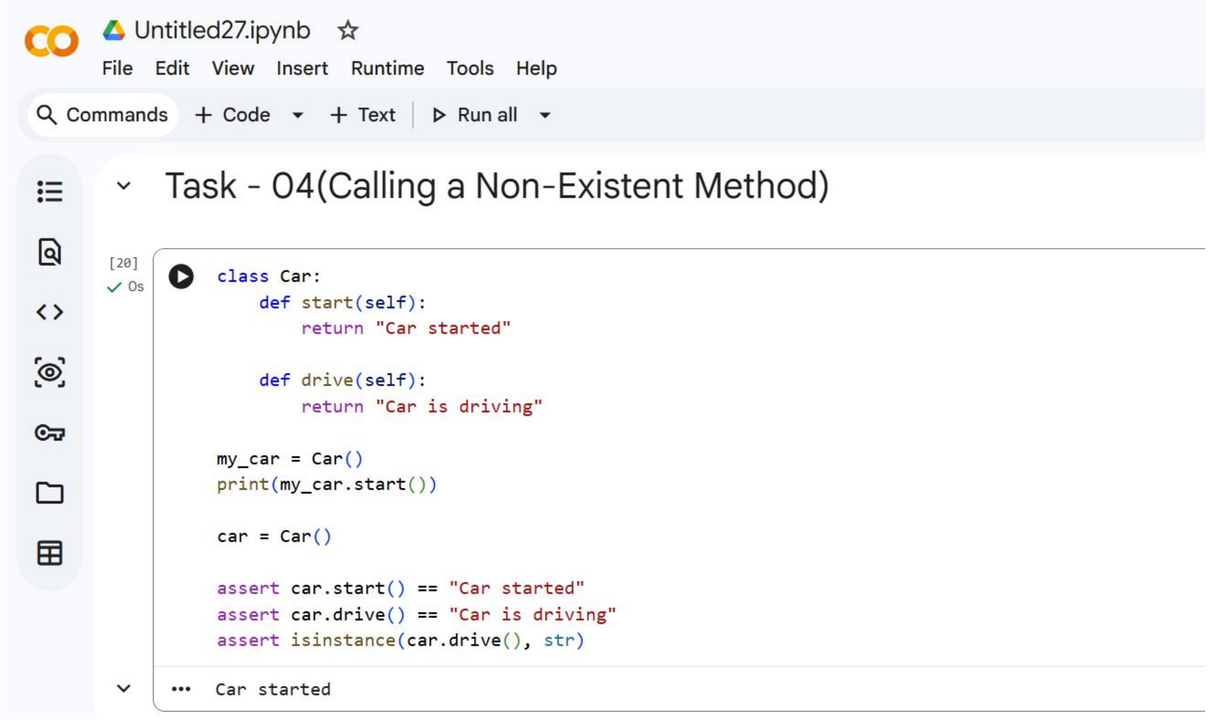
Error:



Please explain this error:

```
AttributeError: 'Car' object has no attribute 'drive'
```

Output:



Task - 04(Calling a Non-Existent Method)

```
class Car:
    def start(self):
        return "Car started"

    def drive(self):
        return "Car is driving"

my_car = Car()
print(my_car.start())

car = Car()

assert car.start() == "Car started"
assert car.drive() == "Car is driving"
assert isinstance(car.drive(), str)
```

```
Car started
```

#Task – 05(TypeError – Mixing Strings and Integers inAddition)

Buggy Code:

```
def add_five(value):

return value + 5

print(add_five("10"))
```

AI Prompt:

The function below throws a TypeError because it mixes a string and an integer.

AI Explanation:

You're encountering a TypeError because you're trying to add a number to a string without converting the string to a number first. The add_five function receives "10" (a string) and tries to add 5 (an integer) to it. To fix this, you need to convert the value to an integer inside the function. I will modify the code to cast the input value to an int.

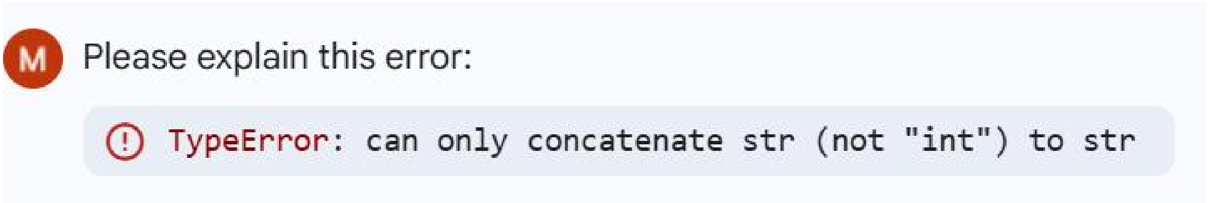
Corrected Code:

```
def add_five(value):

    return int(value) + 5

print(add_five("10"))
```

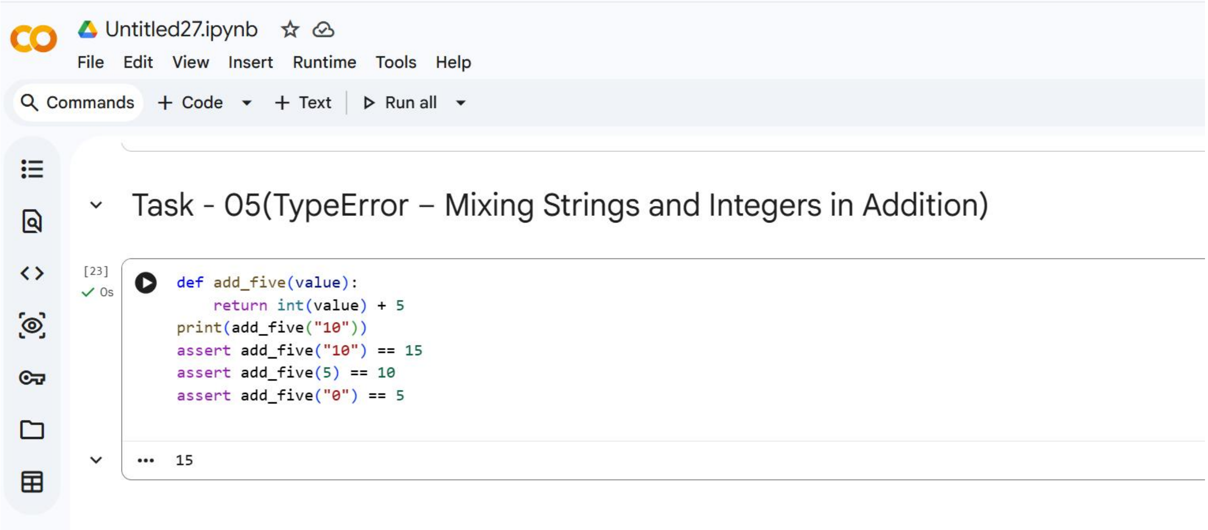
Error:



Please explain this error:

```
TypeError: can only concatenate str (not "int") to str
```

Output:



Untitled27.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[23]
0s

Task - 05(TypeError – Mixing Strings and Integers in Addition)

```
def add_five(value):
    return int(value) + 5
print(add_five("10"))
assert add_five("10") == 15
assert add_five(5) == 10
assert add_five("0") == 5
```

```
15
```