2403a51288

M.ABHINAND REDDY

Lab 7: Error Debugging with AI

# Task 1: Syntax Error – Missing Parentheses in Print Statement

## Buggy Code

def greet():  
 print "Hello, AI Debugging Lab!"  
greet()

## Error Explanation

Python 3 requires parentheses for print(). The code fails with SyntaxError.

## Corrected Code

def greet():  
 return "Hello, AI Debugging Lab!"  
  
# Test cases  
assert greet() == "Hello, AI Debugging Lab!"  
assert isinstance(greet(), str)  
assert "AI Debugging" in greet()  
print(greet())

## Output

Hello, AI Debugging Lab!

# Task 2: Logic Error – Incorrect Condition in If Statement

## Buggy Code

def check\_number(n):  
 if n = 10:  
 return "Ten"  
 else:  
 return "Not Ten"

## Error Explanation

Using = instead of == causes SyntaxError. '=' is assignment, not comparison.

## Corrected Code

def check\_number(n):  
 if n == 10:  
 return "Ten"  
 else:  
 return "Not Ten"  
  
# Test cases  
assert check\_number(10) == "Ten"  
assert check\_number(5) == "Not Ten"  
assert check\_number(-10) == "Not Ten"  
print(check\_number(10))

## Output

Ten

# Task 3: 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"))

## Error Explanation

If the file doesn’t exist, FileNotFoundError is raised.

## Corrected Code

def read\_file(filename):  
 try:  
 with open(filename, 'r') as f:  
 return f.read()  
 except FileNotFoundError:  
 return f"Error: File '{filename}' not found."  
 except Exception as e:  
 return f"Error: {str(e)}"  
  
# Test cases  
assert "not found" in read\_file("nonexistent.txt")  
assert isinstance(read\_file("nonexistent.txt"), str)  
assert read\_file("invalid\_path/abc.txt").startswith("Error")

## Output

Error: File 'nonexistent.txt' not found.

# Task 4: AttributeError – 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

## Error Explanation

drive() is not defined. Either call start() or define drive().

## Corrected Code

class Car:  
 def start(self):  
 return "Car started"  
 def drive(self):  
 return "Car is driving"  
  
my\_car = Car()  
  
# Test cases  
assert my\_car.start() == "Car started"  
assert my\_car.drive() == "Car is driving"  
assert isinstance(my\_car.start(), str)  
print(my\_car.start(), "and", my\_car.drive())

## Output

Car started and Car is driving

# Task 5: TypeError – Mixing Strings and Integers

## Buggy Code

def add\_five(value):  
 return value + 5  
  
print(add\_five("10"))

## Error Explanation

Python does not allow adding str and int. Fix by casting or concatenation.

## Corrected Code

# Solution 1: Type casting  
def add\_five(value):  
 return int(value) + 5  
  
# Test cases  
assert add\_five("10") == 15  
assert add\_five(20) == 25  
assert add\_five(0) == 5  
print(add\_five("10"))  
  
# Solution 2: String concatenation  
def add\_five(value):  
 return str(value) + "5"  
  
# Test cases  
assert add\_five("10") == "105"  
assert add\_five(20) == "205"  
assert add\_five("AI") == "AI5"  
print(add\_five("10"))

## Output

15  
105