Assignment-7

Hall tickets :2303A54038
Batch :48

TASK-1

Prompt : Provide a snippet with a missing parenthesis in print statement(e.g:Print "Hello") use AI detect and fix the syntax error

Code :

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

Fixing the code :

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

Proper Explanation

In the original code, the print statement is missing parentheses, which is a syntax error in Python 3. The correct syntax for the print function requires parentheses around the string to be printed. In the fixed code, we added parentheses around the string "Hello, AI Debugging Lab!" to ensure that it is properly executed as a function call.

TASK-2

Prompt :

Supply a function where an if- condition mistakenly uses = indtead of == .let Ai identify and fix the issue

Code:

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

Fixing the code :

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

Proper Explanation:

In the original code, the if condition uses a single equals sign (=), which is an assignment operator, instead of a double equals sign (==), which is a comparison operator. This will result in a syntax error. In the fixed code, we replaced the single equals sign with a double equals sign to correctly compare the value of n to 10.

TASK-3:

Prompt:

provide code that attempts to open a non-existemt file and crashes .use AI to apply safe error handling

Code:

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

Fixing the code :

```
def read_file(filename):
    try:
        with open(filename, 'r') as
            return f.read()
    except FileNotFoundError:
        return f"Error: The file '{
print(read_file("nonexistent.txt"))
```

Proper Explanation:

In the original code, if the specified file does not exist, the program will raise a FileNotFoundError and crash. In the fixed code, we wrap the file opening operation in a try-except block to catch the FileNotFoundError. If the error occurs, we return a user-friendly error message instead of allowing the program to crash.

TASK-4

Prompt :

Give a class where a non -existent method is called(eg:obj.undefined_method()).use AI to debug and fix

Code:

```
class Car:
    def start(self):
```

```
        return "Car s
my_car = Car()
print(my_car.start())
```
Fixing the code :
```
class Car:
    def start(self):
        return "Car s
my_car = Car()
print(my_car.start())
```
Explanation :
In the original code, the method name "startedd" is misspelled, which would lead to an AttributeError when trying to call my_car.start(). In the fixed code, we corrected the method name to "start" to ensure that it can be called successfully without any errors.

TASK -5

Prompt:
provide code that adds an integer and string("5"+2) causing a typeerror .use Ai to resolve the bug

Code:
```
def add_five(value):
    return value + 5
print(add_five("10"))
```
Fixing the code :
```
def add_five(value):
    try:
        return int(va
    except ValueError
        return "Error
print(add_five("10"))
```
Explanation :
In the original code, the function add_five attempts to add an integer (5) to a string ("10"), which results in a TypeError. In the fixed code, we convert the input value to an integer using int(value) before performing the addition. We also added error handling to catch any ValueError that may occur if the input cannot be converted to an integer, providing a user-friendly error message instead.