**AI ASSISTED CODING**

**LAB ASSIGNMENT -7.2**

Program : B-TECH

Specialization : AIML

Name of Student : DONIKELA LAXMAN

Enrollment-No : 2403A51283s

Batch No : 01

Date : 16-09-2025

**TASK DESCRIPTION-1**

**Task -1 –** Syntax Error in Conditionals

PROMPT:

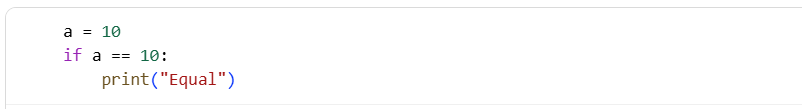
Correct the syntactical error from the code

a=10

if a = 10:

print("Equal")

GENERATED CODE:

****

Output:-

****

OBSERVATION:

The code is simple, and it can be done by manually also.but the code generated is just a basic bython syntax rule and it is corrected by the gemini simply**.**

**2)**

Task -2:

Loop Off-By-One Error.

PROMPT :

Correct the mistake in the code, loop-off-by-one Error.

def sum\_upto\_n(n):

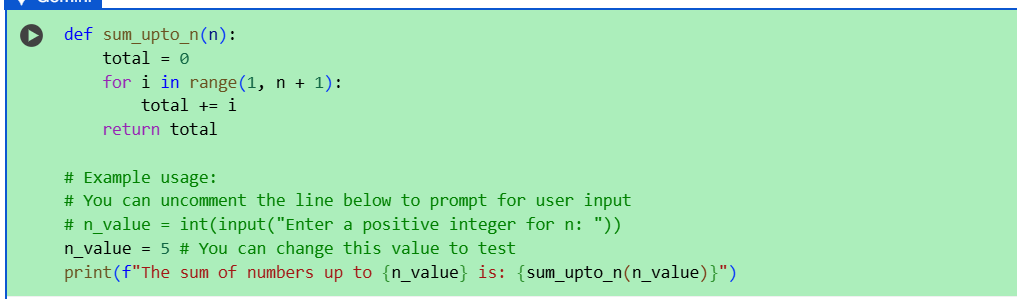
total=0

for i in range(1,n):

total +=i

return total. The n value should be user-defined

CODE:

****

OUTPUT:

****

OBSERVATION:

it's just a matter of small changes, but it can change the entire code.I asked it to identify the error, and it gave me the correct code. It provided the working code after debugging.

**3) TASK-3**

Error: AttributeError

PROMPT:

Fix the increment and decrement errors in the code.

class User:

def \_\_init\_\_(self, name):

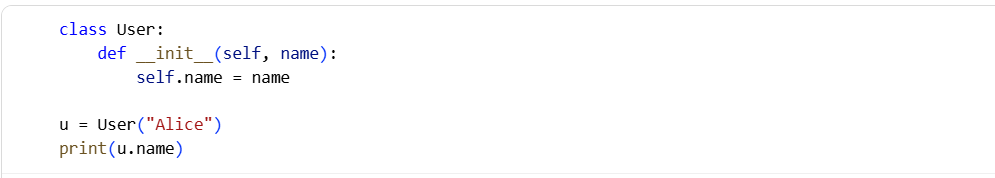
self.name=name

u = User("Alice")

print(u.getName()).

And the error is an attribute error

CODE:

****

OUTPUT:

****

OBSERVATION:

I see the AttributeError. The User class doesn't have a method called getName(). You can access the name attribute directly using dot notation. Also, there are some indentation issues in the provided code. Here is the corrected code:

Excellent! The corrected code ran without errors and printed the user's name.

**4)**

TASK-4:

Incorrect Class Attribute Initialization.

PROMPT:

correct the incorrect attribute initialization

class Car:

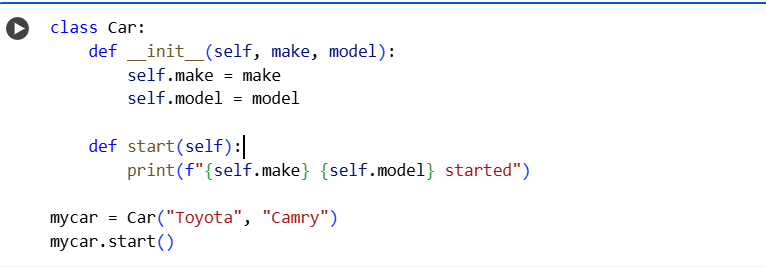
def start():

print("car started")

mycar = Car()

mycar.start().detect the missing self and initialize attributes properly.

GENERATED CODE:

****

**OUTPUT:**

****

**OBSERVATION:**Car: def start(): print("car started") mycar = Car() mycar.start() Detect the missing self and initialize attributes properly.

You're right, the start method is missing the self parameter. Also, to initialize attributes like make and model, we need to add an \_\_init\_\_ method. Here is the corrected code with proper attribute initialization and the self parameter in the start method**.**

**5)**

**TASK-5:**

Conditional Logic Error in Grading System

PROMPT:

correct the conditional Logic Error in grading system from the the code

def grade\_student(score):

if score < 40:

return "A"

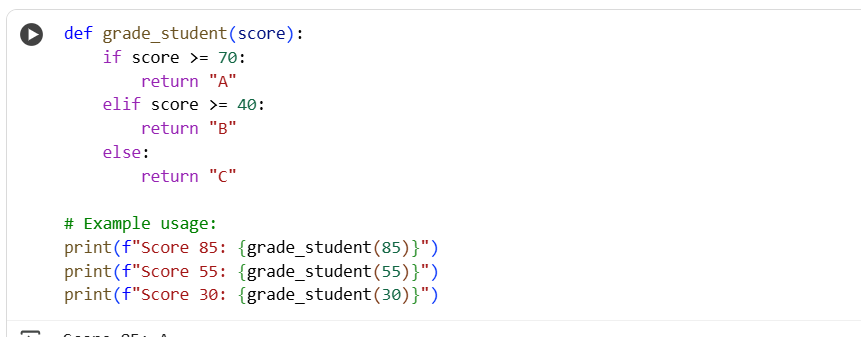
elif score < 70:

return "B"

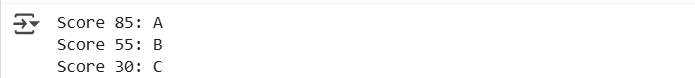
else:

return "c"

**GENERATED CODE:**

****

**OUTPUT:**

****

**OBSERVATION:**

I see the conditional logic error in the grading system. The current logic has overlapping conditions and doesn't represent a typical grading scale. I'll correct it to a more standard system where:

* Scores 70 and above get an "A"
* Scores between 40 and 69 get a "B"
* Scores below 40 get a "C"

**THANKYOU**