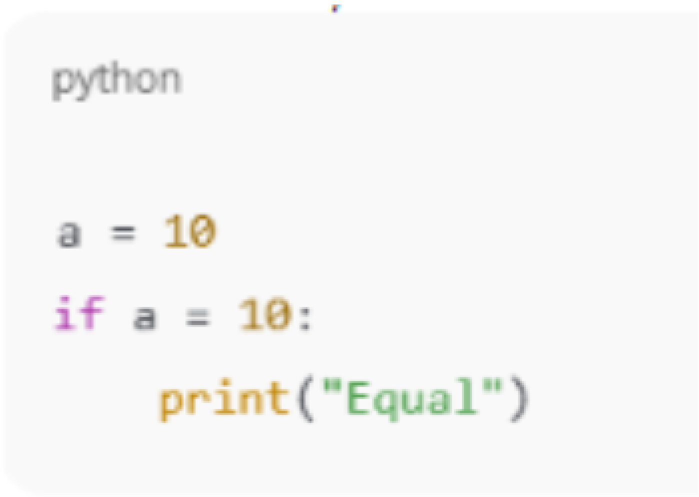
AI ASSISTED CODING

LAB ASSIGNMENT -7.2

Program : B-TECH
Specialization : AIML
Name of Student : AKULA MANOJ
Enrollment-No : 2403A52031
Batch No : 02
Date : 16-09-2025

TASK DESCRIPTION-1

Task -1 – Syntax Error in Conditionals



python

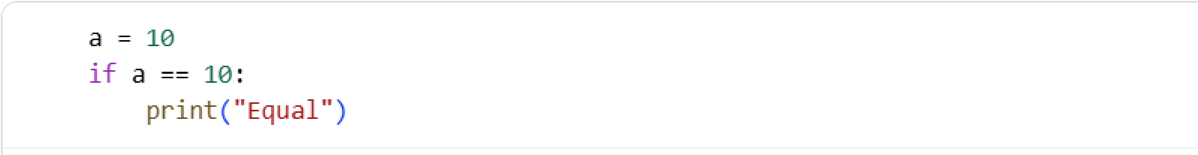
```python
a = 10
if a = 10:
    print("Equal")
```

PROMPT:

Correct the syntactical error from the code

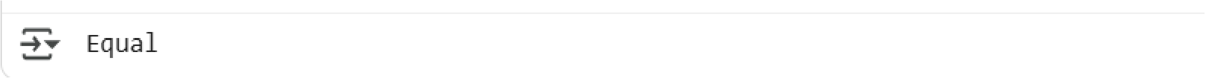
```
a=10

if a = 10:

print("Equal")
```

GENERATED CODE:



```
a = 10
if a == 10:
    print("Equal")
```

Output:-



```
Equal
```

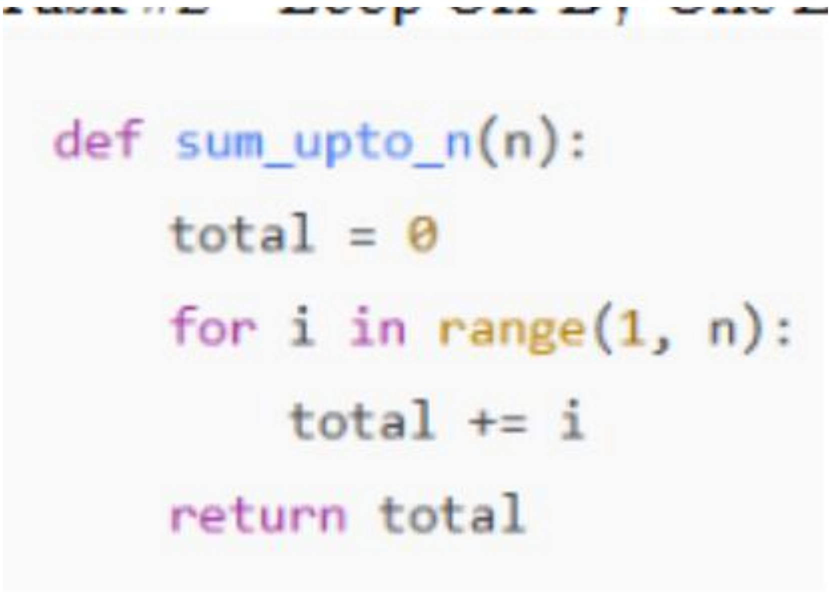
OBSERVATION:

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.



```
def sum_upto_n(n):
    total = 0
    for i in range(1, n):
        total += i
    return total
```

PROMPT :

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

def sum_upto_n(n):

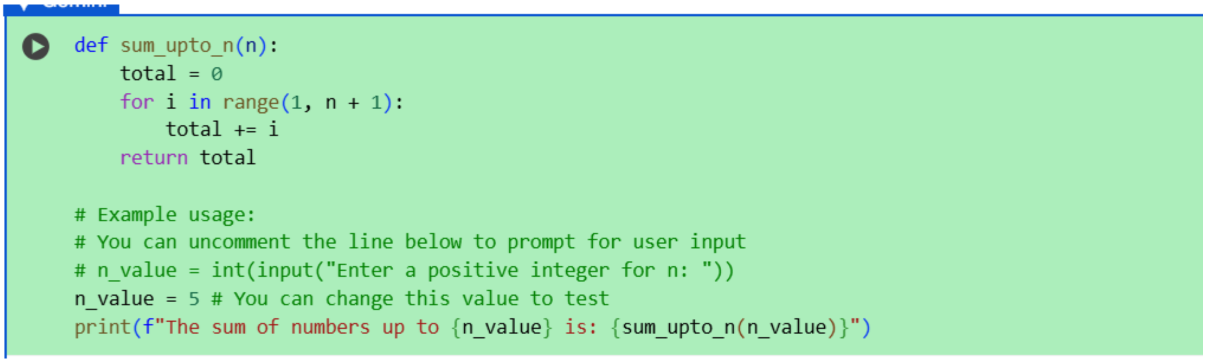
  total=0

for i in range(1,n):

total +=i

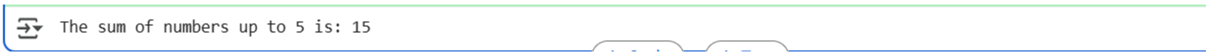
return total. The n value should be user-defined

CODE:



```python
def sum_upto_n(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total

# Example usage:
# You can uncomment the line below to prompt for user input
# n_value = int(input("Enter a positive integer for n: "))
n_value = 5 # You can change this value to test
print(f"The sum of numbers up to {n_value} is: {sum_upto_n(n_value)}")
```

OUTPUT:



```
The sum of numbers up to 5 is: 15
```

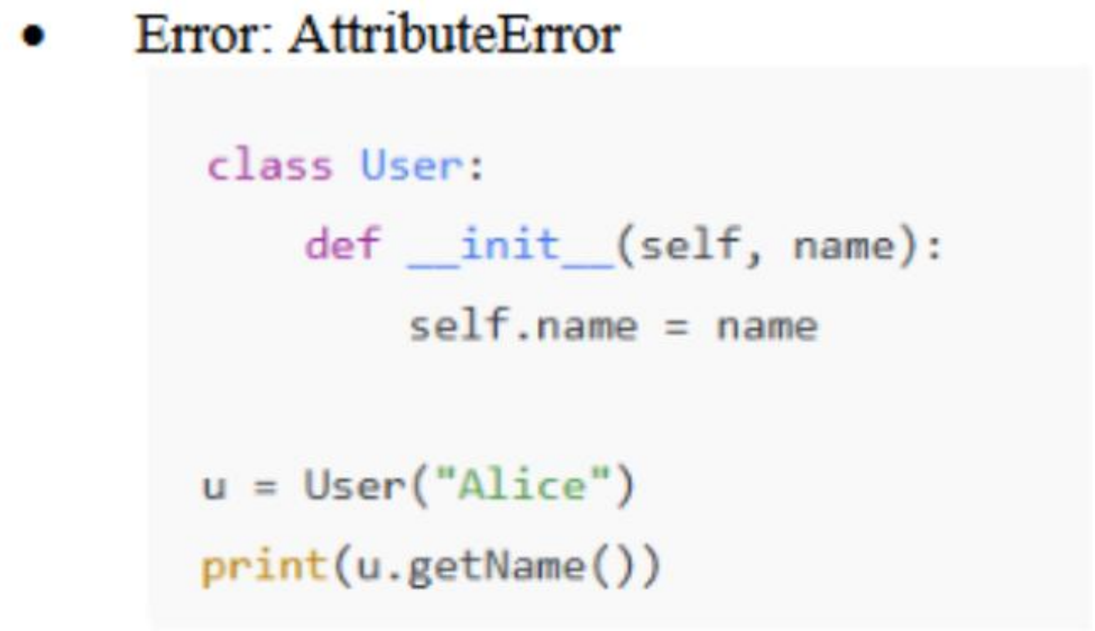
OBSERVATION:

I solved this by using Google Colab's Gemini, and I wrote the debugged code manually. There is no large debugging in the code; 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

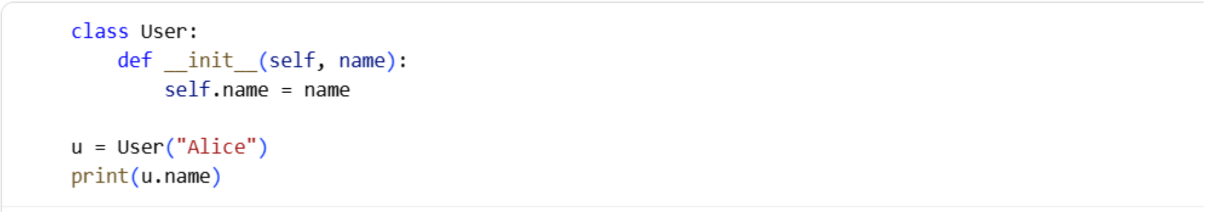
Error: AttributeError



- Error: AttributeError

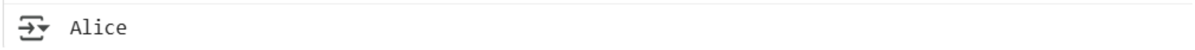
```python
class User:
    def __init__(self, name):
        self.name = name

u = User("Alice")
print(u.getName())
```

CODE:



```python
class User:
    def __init__(self, name):
        self.name = name

u = User("Alice")
print(u.name)
```

OUTPUT:



```
Alice
```

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.

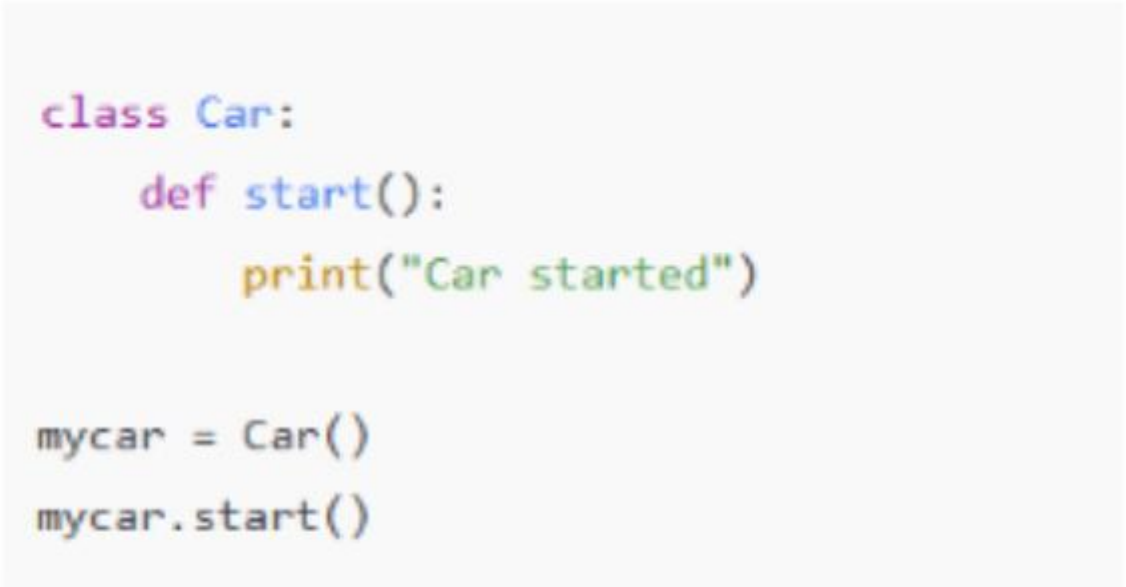
Also, there are some indentation issues in the provided code. Here is the corrected code:

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

4)

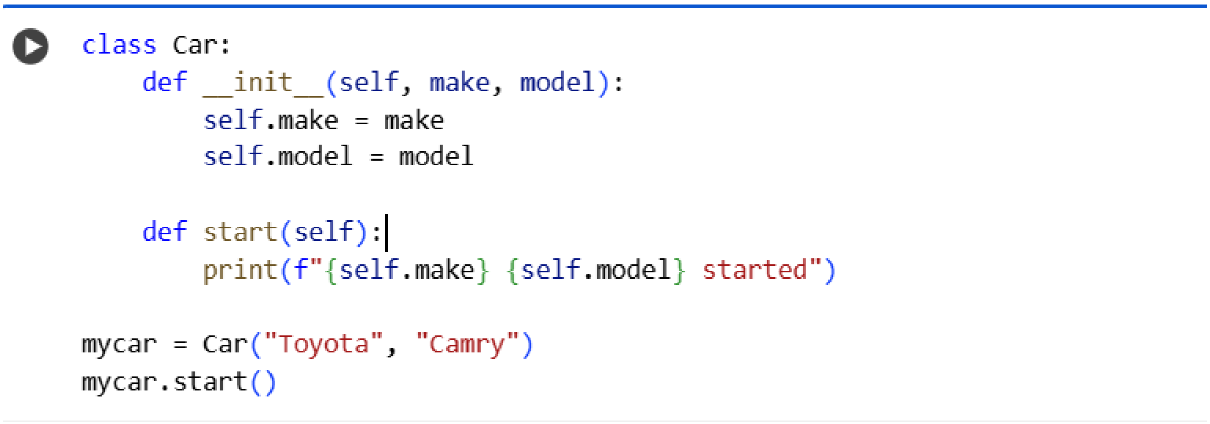
TASK-4:

Incorrect Class Attribute Initialization.



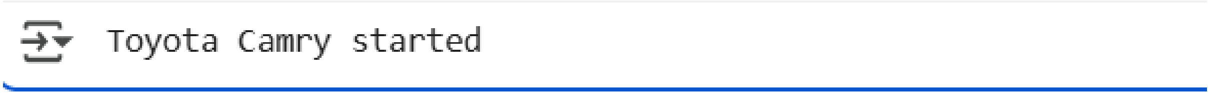
```
class Car:
    def start():
        print("Car started")

mycar = Car()
mycar.start()
```

GENERATED CODE:



```
class Car:
    def __init__(self, make, model):
        self.make = make
        self.model = model

    def start(self):
        print(f"{self.make} {self.model} started")

mycar = Car("Toyota", "Camry")
mycar.start()
```

OUTPUT:



```
Toyota Camry started
```

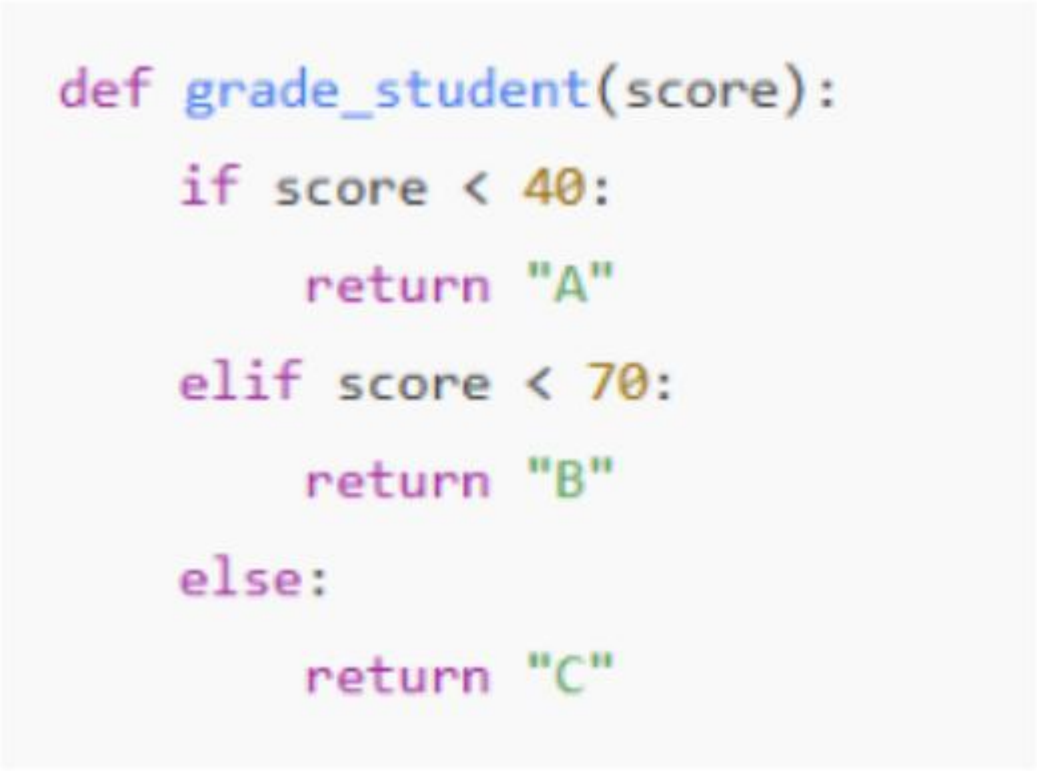
OBSERVATION:

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)

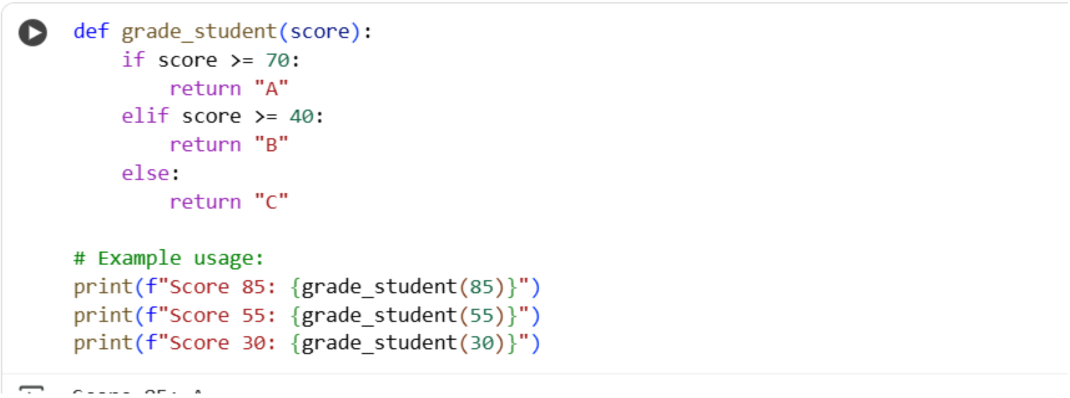
TASK-5:



```
def grade_student(score):
    if score < 40:
        return "A"
    elif score < 70:
        return "B"
    else:
        return "C"
```

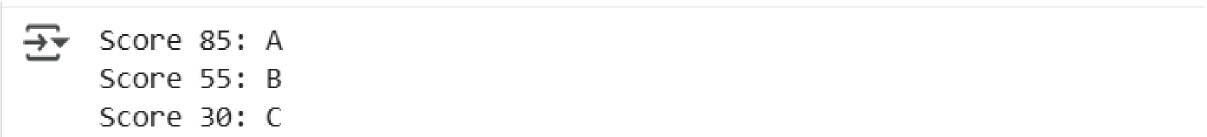
Conditional Logic Error in Grading System

GENERATED CODE:



```
def grade_student(score):
    if score >= 70:
        return "A"
    elif score >= 40:
        return "B"
    else:
        return "C"

# Example usage:
print(f"Score 85: {grade_student(85)}")
print(f"Score 55: {grade_student(55)}")
print(f"Score 30: {grade_student(30)}")
```

OUTPUT:



```
Score 85: A
Score 55: B
Score 30: C
```

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