PROGRAM: B.TECH

SPECIALIZATION: CSE - AIML

COURSE TITLE: AI CODING

SEMESTER : 3RD SEM

NAME OF STUDENT: SRIYA

ENROLLMENT NO: 2403A51350

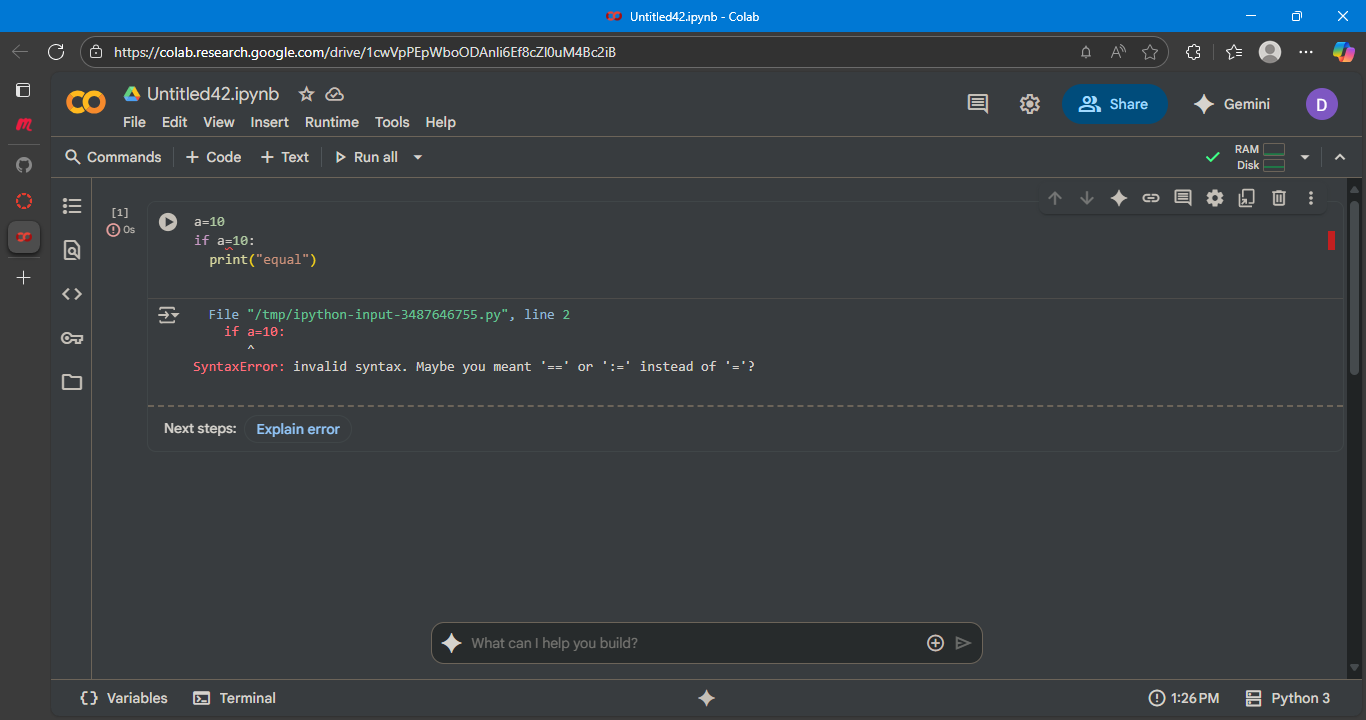
BATCH NO: 01

Questions:

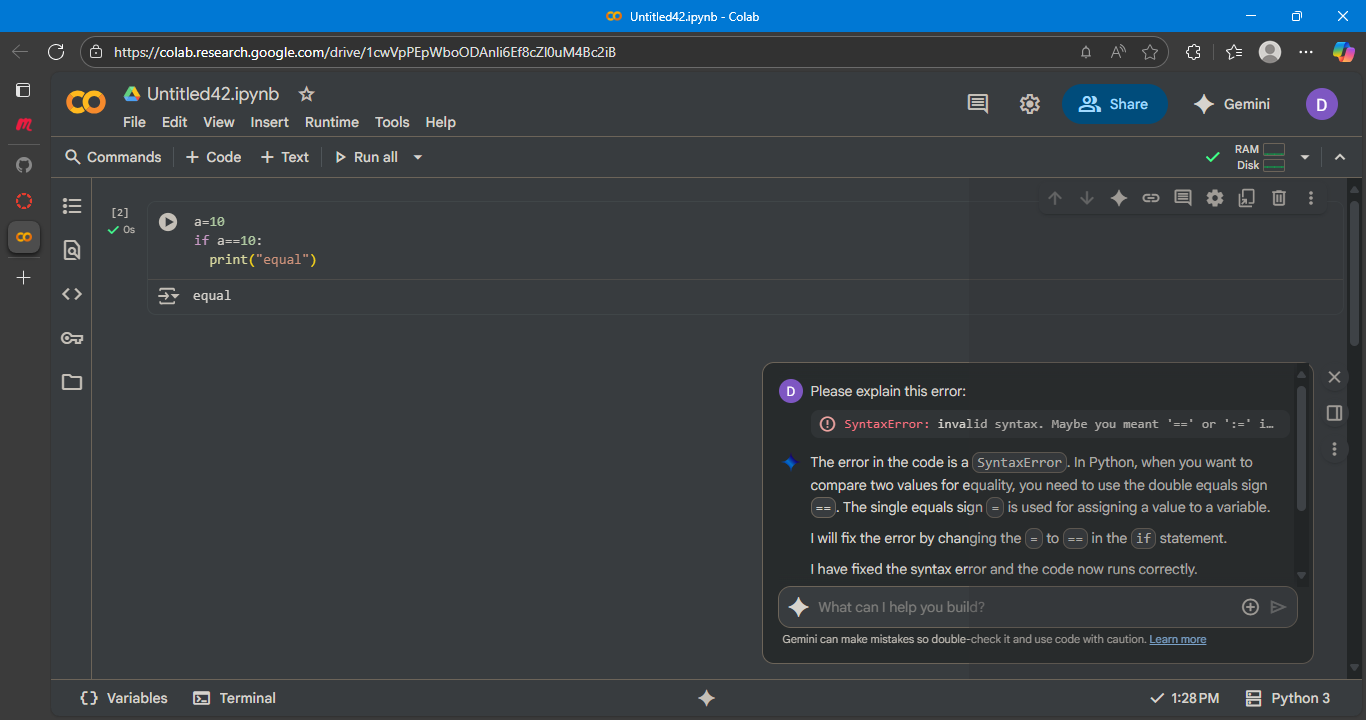
Task Description#1  
• Task #1 – Syntax Error in Conditionals

Expected Output#1  
• Corrected function with syntax fix

Error:



With corrected code:



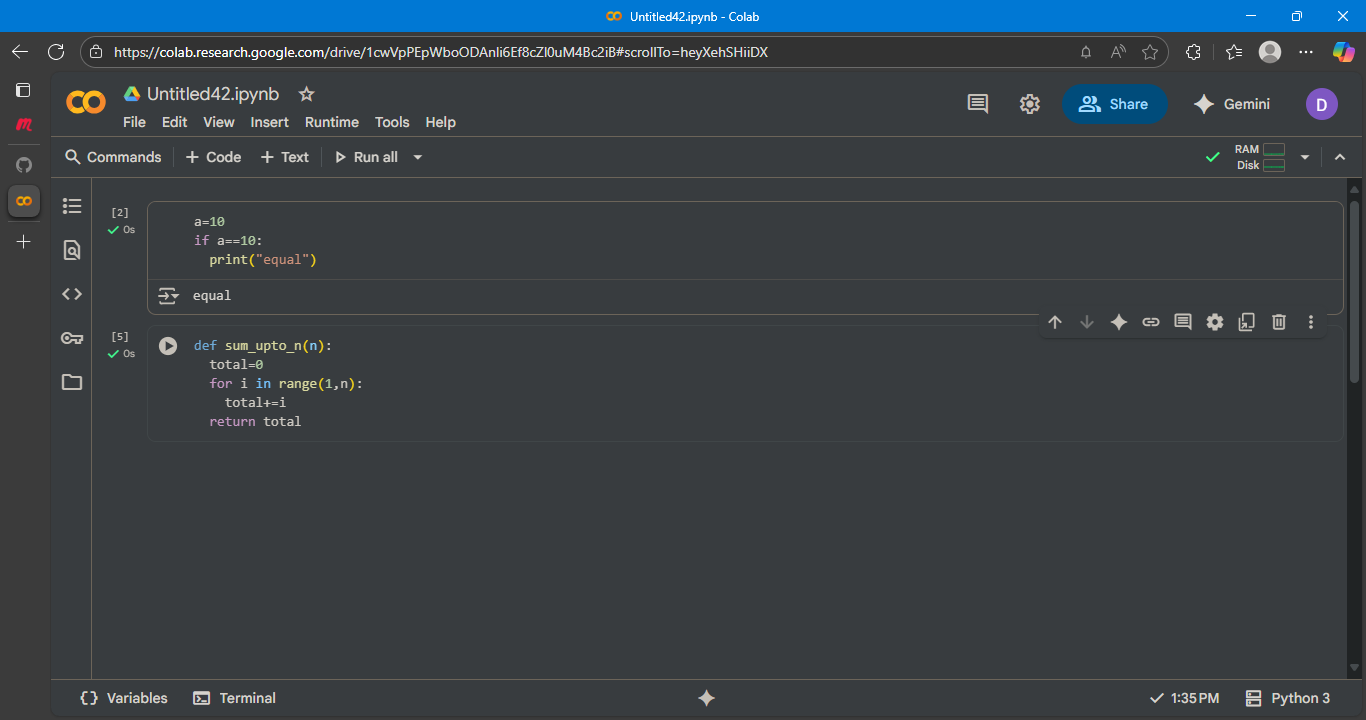
Explanation:  
Certainly! This code snippet does the following:

1. a = 10: This line assigns the integer value 10 to the variable a.
2. (if a == 10: ) (: This is a conditional statement. It checks if the value of the variable a is equal to 10. The == operator is used for comparison.)
3. print("equal"): If the condition in the if statement is true (i.e., if a is indeed equal to 10), the code inside this block is executed, which is printing the string "equal" to the console.

Since a is assigned the value 10, the condition a == 10 is true, and the code prints "equal".

Task Description#2 (Loops)  
• Task #2 – Loop Off-By-One Error.

Expected Output#2  
• AI fixes increment/decrement error



Prompt 1:

explain the error in the code?

The code in cell  assigns the value 10 to variable a, and then checks if a is equal to 10 and prints "equal". The code in cell  defines a function sum\_upto\_n that calculates the sum of numbers from 1 up to (but not including) n.

If you were expecting an error, could you please specify which cell you were referring to or describe the error you encountered?

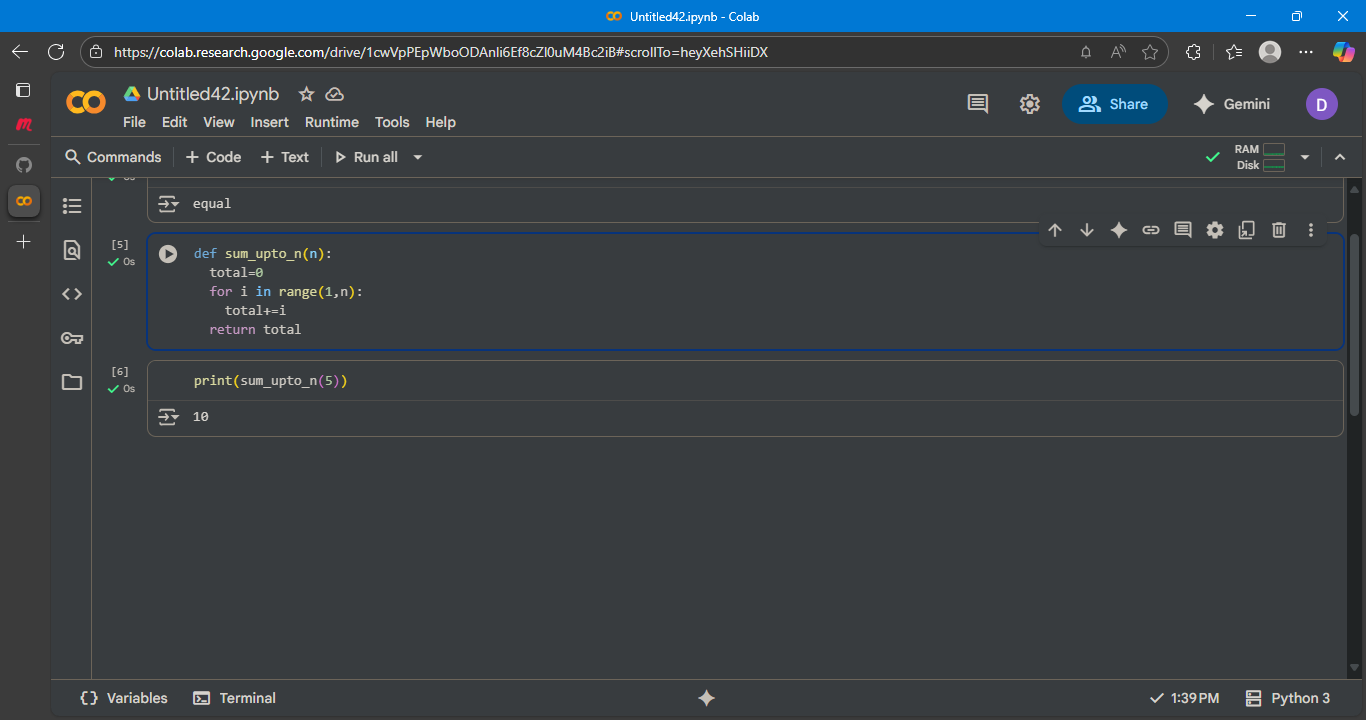
Prompt 2:

explain the error in the code why it's not generating output?

I see the confusion! The code in cell heyXehSHiiDX defines a function called sum\_upto\_n, but it doesn't actually call the function. Defining a function just tells the program what to do when the function is called; it doesn't execute the code inside the function immediately.

To see output, you need to call the function with an argument. For example, to see the sum of numbers up to 5 (excluding 5), you would call sum\_upto\_n(5) and print the result.

Corrected code output:



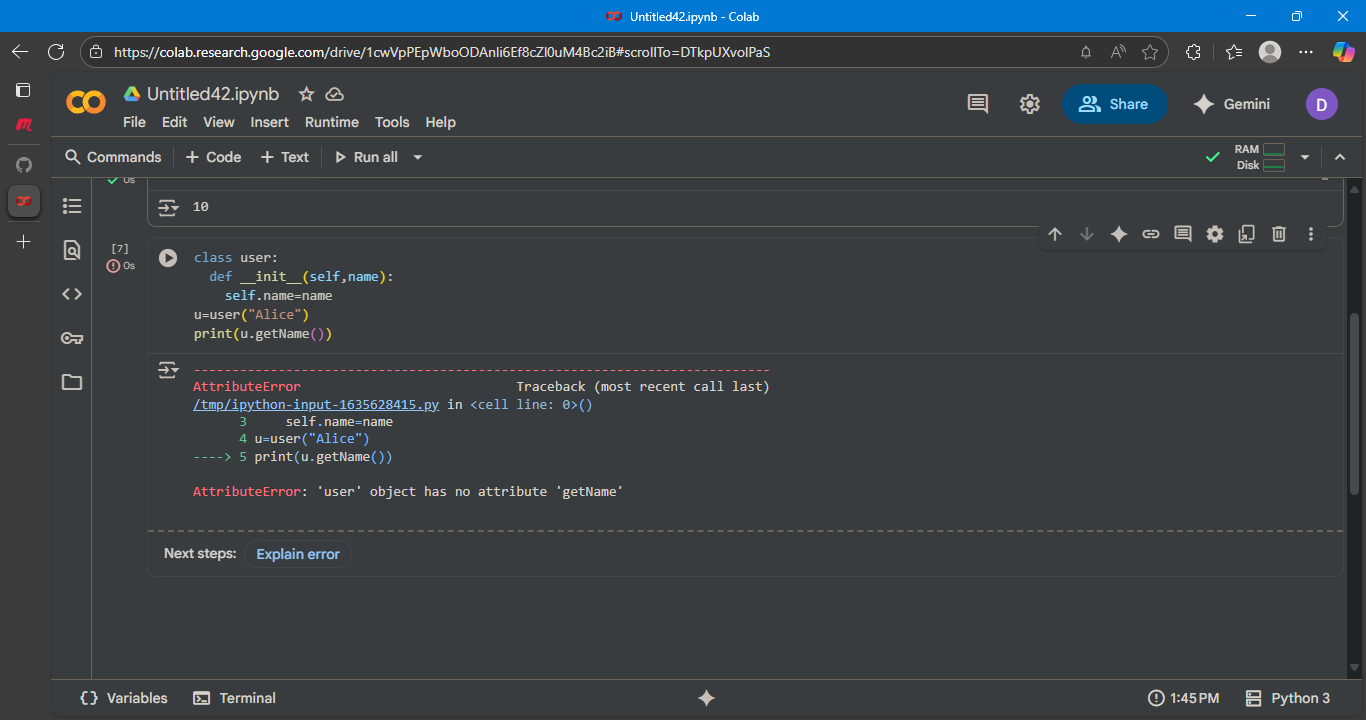
Explanation:

1. sum\_upto\_n(5): This part calls the function sum\_upto\_n and passes the value 5 as an argument to the parameter n. The function then calculates the sum of numbers from 1 up to (but not including) 5, which is 1 + 2 + 3 + 4 = 10. The function returns this value (10).
2. print(...): This part takes the value returned by sum\_upto\_n(5) (which is 10) and prints it to the console.

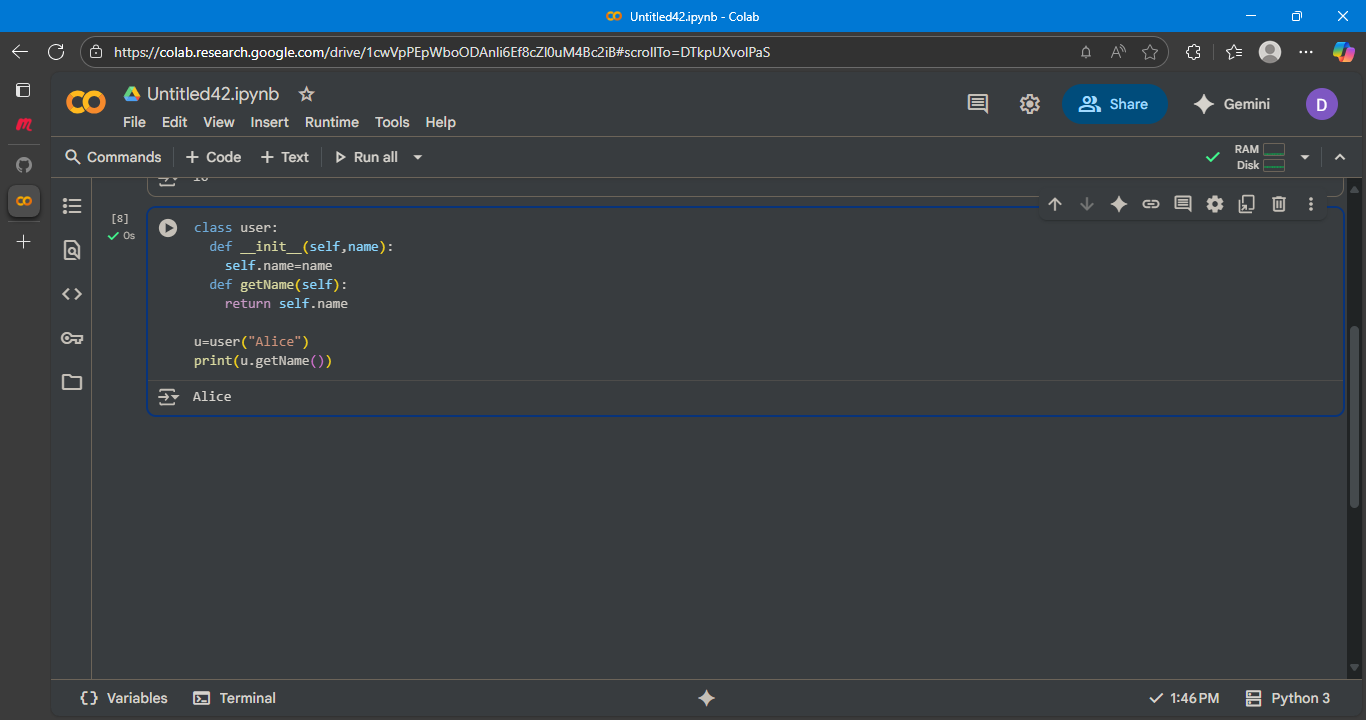
Task Description#3  
• Error: Attribute Error

Expected Output#3  
• Identify the missing method and correct the code.

Error:



Corrected code:

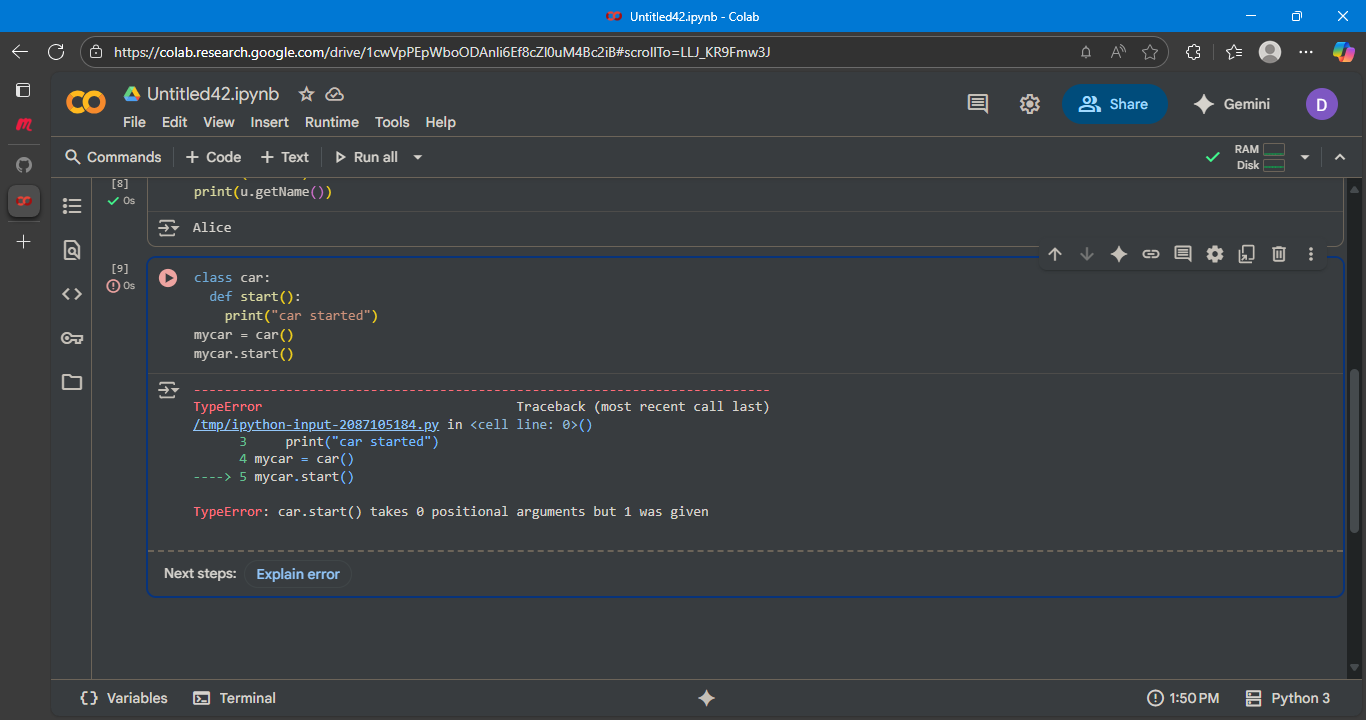


Explanation:

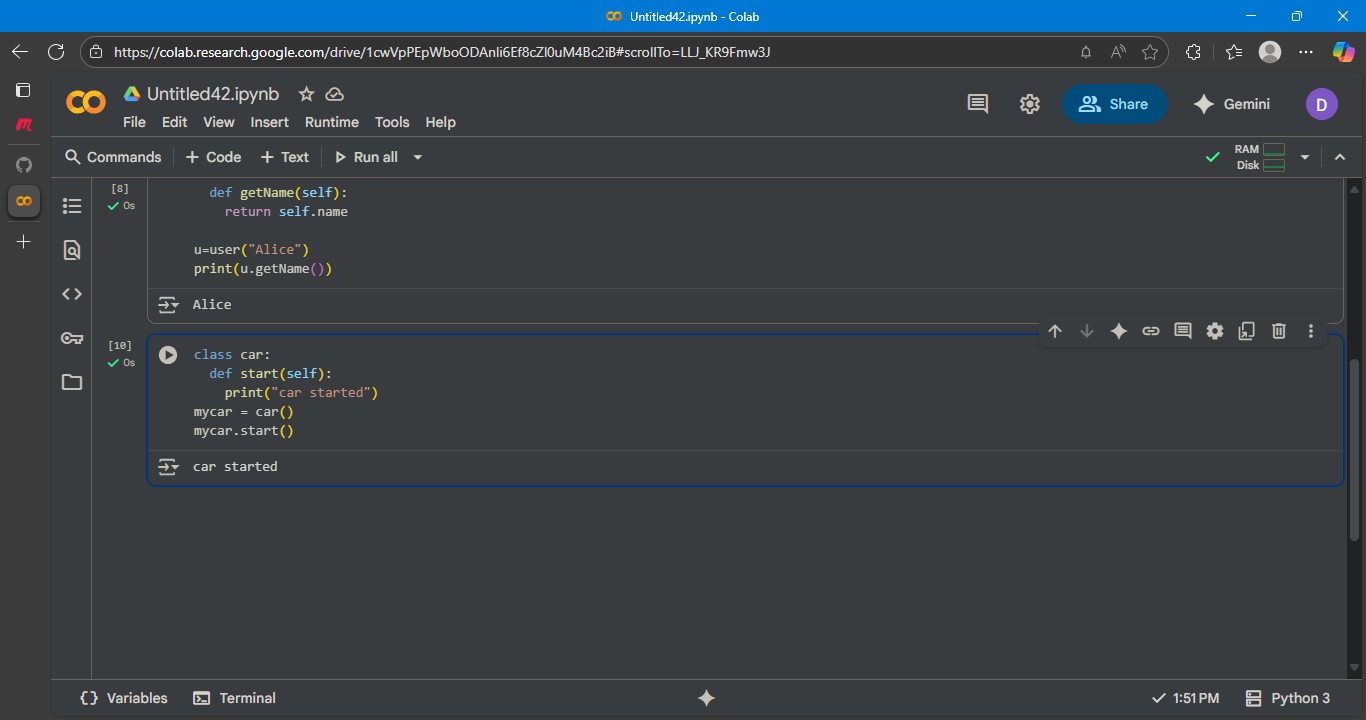
1. **class user:**: This line defines a new class named user. Classes are blueprints for creating objects.
2. **def \_\_init\_\_(self, name):**: This is the constructor method of the class. It's called automatically when you create a new user object.
   * self: Refers to the instance of the class being created.
   * name: This is a parameter that you pass when you create a user object.
   * self.name = name: This line takes the name value that was passed in and assigns it to an attribute called name within the object (self.name).
3. **def getName(self):**: This defines a method within the user class called getName.
   * self: Again, refers to the instance of the class.
   * return self.name: This method simply returns the value of the name attribute of the object.
4. **u = user("Alice")**: This line creates a new instance (an object) of the user class. It calls the \_\_init\_\_ method and passes the string "Alice" as the name argument. This creates a user object named u with its name attribute set to "Alice".
5. **print(u.getName())**: This line calls the getName() method on the u object and then prints the value returned by that method, which is "Alice"

Task Description#4  
• Incorrect Class Attribute Initialization  
Expected Output#4  
• Detect missing self and initialize attributes properly

Error:



Corrected code:



Explanation:

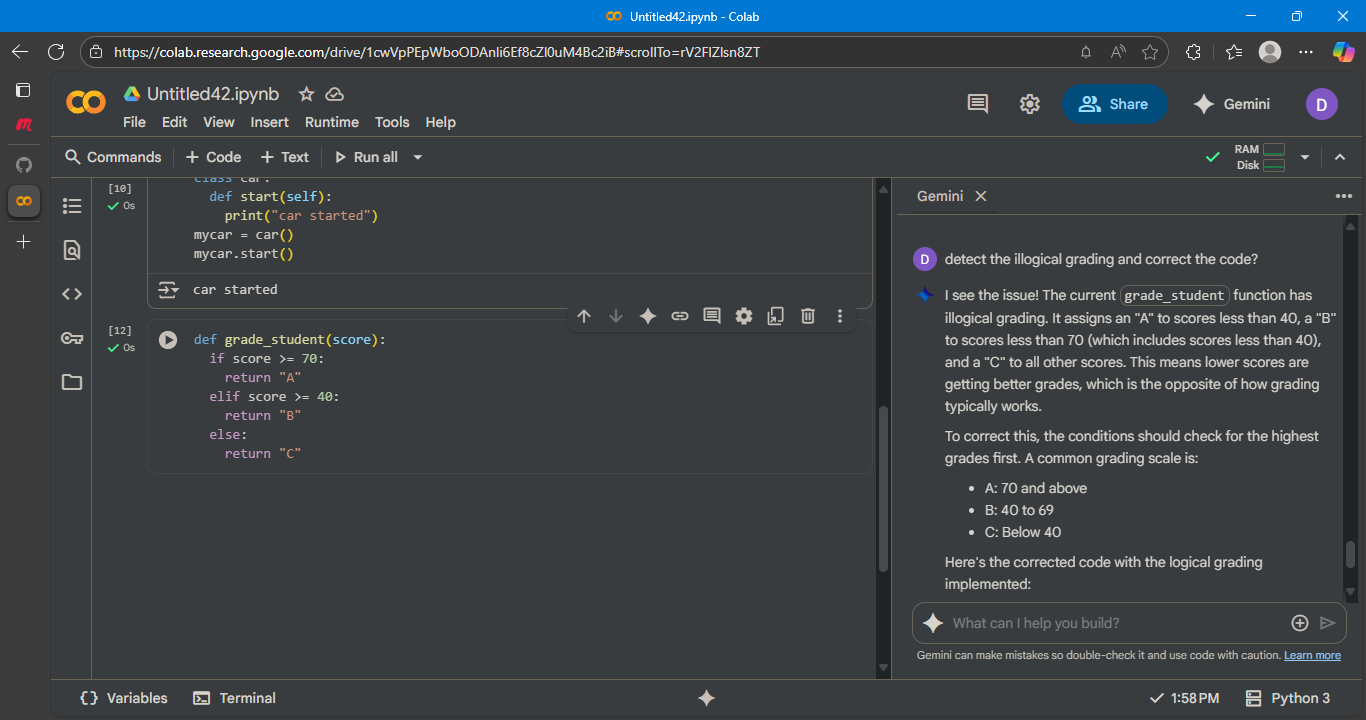
1. **class car:**: This line defines a new class called car. This is a blueprint for creating "car" objects.
2. **def start(self):**: This defines a method within the car class named start.
   * self: This is a required parameter in instance methods and refers to the instance of the class the method is being called on.
   * print("car started"): When the start method is called on a car object, this line will be executed, printing "car started" to the console.
3. **mycar = car()**: This line creates a new instance (an object) of the car class and assigns it to the variable mycar.
4. **mycar.start()**: This line calls the start() method on the mycar object. This executes the code inside the start method, resulting in "car started" being printed

Task Description#5  
• Conditional Logic Error in Grading System  
Expected Output#5  
• Detect illogical grading and correct the grade levels

Prompt:

detect the illogical grading and correct the code?

Corrected code:



Explanation:

1. **def grade\_student(score):**: This line defines a function named grade\_student that accepts one argument, score.
2. **if score >= 70:**: This is the first condition. If the score is greater than or equal to 70, the code inside this block is executed.
3. **return "A"**: If the first condition is true, the function immediately stops and returns the string "A".
4. **elif score >= 40:**: This is the second condition, checked only if the first condition is false. If the score is greater than or equal to 40, the code inside this block is executed.
5. **return "B"**: If the second condition is true, the function immediately stops and returns the string "B".
6. **else:**: This block is executed if neither of the previous conditions is true.
7. **return "C"**: If the score is less than 40, the function returns the string "C".