**ASSIGNMENT – 7.3**

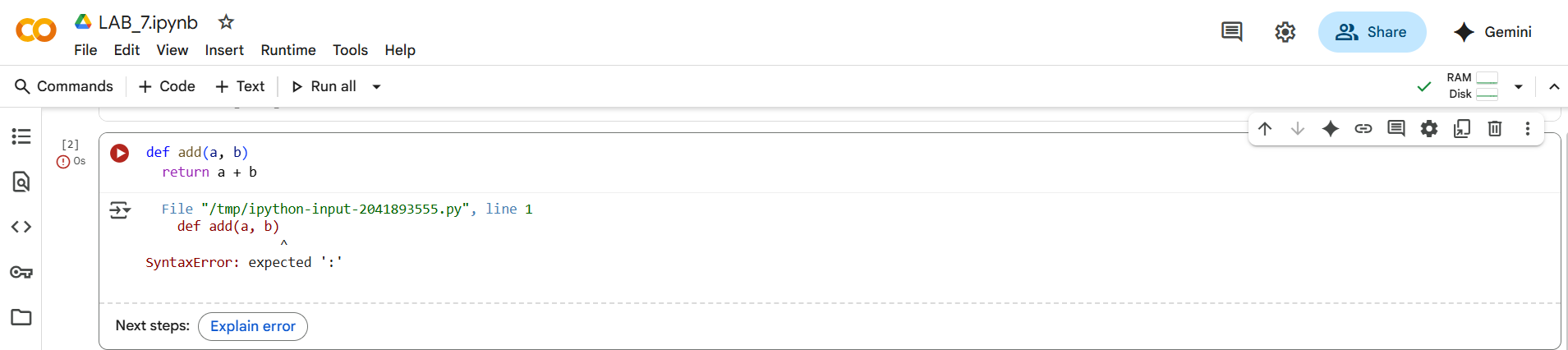
**NAME : SAI CHARAN .P**

**ROLL:NO : 2403A52343**

**BATCH : AI 13**

**TASK-1:** Syntax Error

**Wrong code:**

****

**Corrected Code:**

**A close up of a text

AI-generated content may be incorrect.**

**Error Explanation:**

The code defines a Python function called add that takes two arguments, a and b. Inside the function, it calculates the sum of a and b with the expression a + b. There is also a comment # Added a colon here to fix the SyntaxError which indicates that a colon was added at the end of the def line to correct a syntax error.

However, the function currently does not have a return statement. This means that while the addition a + b is performed internally when the function is called, the result of this calculation is not sent back as the output of the function. If you call this function, it will execute without error, but it won't produce a value that you can use or print.

To make the function return the sum of a and b, you need to add return a + b inside the function.

**TASK-2:** Logic Error

**Wrong Code:**

**A white board with text and numbers

AI-generated content may be incorrect.**

**Corrected Code:**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

**Error Explanation:**

This code defines a Python function called count\_down that takes one argument, n.

* def count\_down(n): This line defines the function named count\_down which accepts one parameter n.
* while n >= 0: This is a while loop that continues to execute as long as the value of n is greater than or equal to 0.
* print(n): Inside the loop, this line prints the current value of n.
* n -= 1: This line decrements the value of n by 1 in each iteration of the loop.
* # Corrected from n += 1 to make the function count down and terminate: This is a comment explaining that this line was changed from n += 1 (which would cause an infinite loop) to n -= 1 to ensure the function counts down and eventually stops when n becomes less than 0.

So, when you call this function with a non-negative integer, it will print that number and then each subsequent smaller integer until it reaches 0, and then the loop will terminate.

**TASK-3:** Division by Zero Error Handling

**Wrong Code:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Correct Code:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Error Explanation:**

This code defines a Python function called divide that takes two arguments, a and b, and performs division while handling the case of division by zero.

* def divide(a, b):: This line defines the function named divide which accepts two parameters, a (the numerator) and b (the denominator).
* if b == 0:: This is an if statement that checks if the value of b is equal to 0. This condition is true when you are trying to divide by zero.
* return "Error: Cannot divide by zero.": If the if condition is true (i.e., b is 0), the function immediately returns the string "Error: Cannot divide by zero.". This prevents the ZeroDivisionError from occurring.
* return a / b: If the if condition is false (i.e., b is not 0), this line is executed. It calculates the result of dividing a by b and returns the result.
* print(divide(10, 0)): This line calls the divide function with a set to 10 and b set to 0, and then prints the value returned by the function. Since b is 0, the function returns the error message, which is then printed to the output.

In summary, this code defines asafe division function that checks for division by zero before performing the division, returning an error message in that specific case.

**TASK-4:** Class Definition Error and Correction

**Wrong Code:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Corrected Code:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Error Explanation:**

This code defines a class called Rectangle and creates an object (or instance) of that class.

* class Rectangle: starts the definition of the class blueprint.
* def \_\_init\_\_(self, length, width): is a special method called the constructor. It's run automatically when you create a new Rectangle object.
  + self refers to the newly created object itself.
  + length and width are parameters that will be used to set the dimensions of the rectangle.
* self.length = length and self.width = width store the provided length and width values as attributes of the Rectangle object.
* rect = Rectangle(5, 3) creates a specific Rectangle object with a length of 5 and a width of 3, and assigns it to the variable rect.
* print(rect) prints the default string representation of the rect object, which typically shows the object's type and memory address.

**TASK-5:** List Index Error Handling

**Wrong Code:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Corrected Code:**

**A close-up of a white background

AI-generated content may be incorrect.**

**Error Explanation:**

This code demonstrates how to create a Python list and access its elements using indexing.

* numbers = [1, 2, 3, 4]: This line creates a list named numbers and initializes it with four integer values: 1, 2, 3, and 4.
* print(numbers[0]): This line accesses an element within the numbers list and prints it.
  + numbers: This is the name of the list.
  + [0]: This is the index used to access an element. In Python (and many other programming languages), list indices start from 0. So, [0] refers to the *first* element in the list, [1] refers to the second, and so on.
  + # Accessing a valid index (0): This is a comment explaining that the index 0 is a valid index for this list.

When this code is executed, it will print the element at index 0 of the numbers list, which is the number 1.