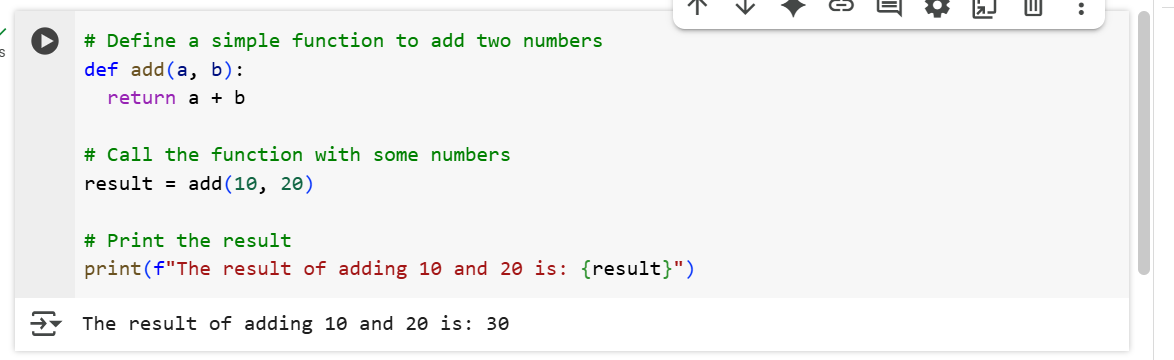
**ASSINGMENT 7**

**H.NO: 2403A52139**

* **TASK-1**

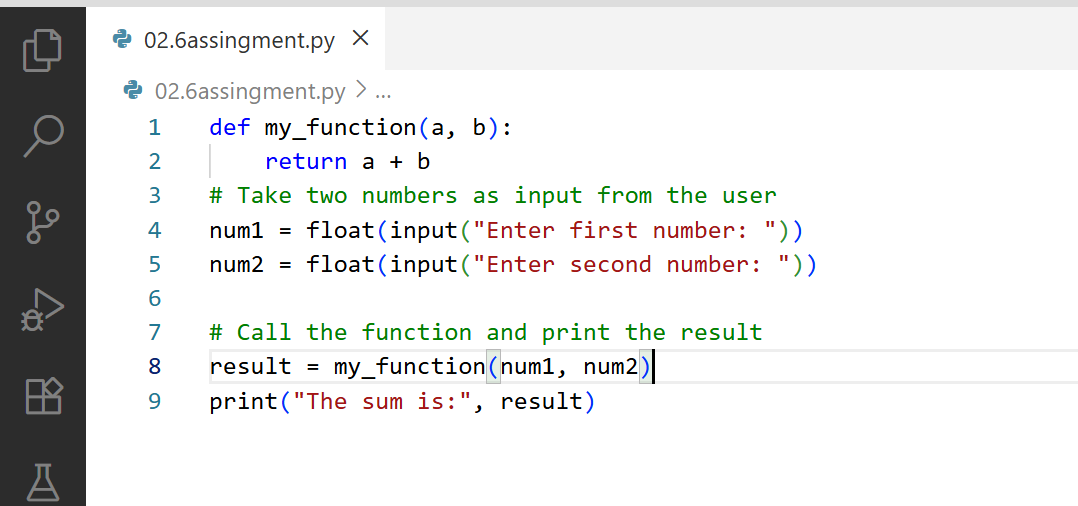
Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.

**Gemini code and output**

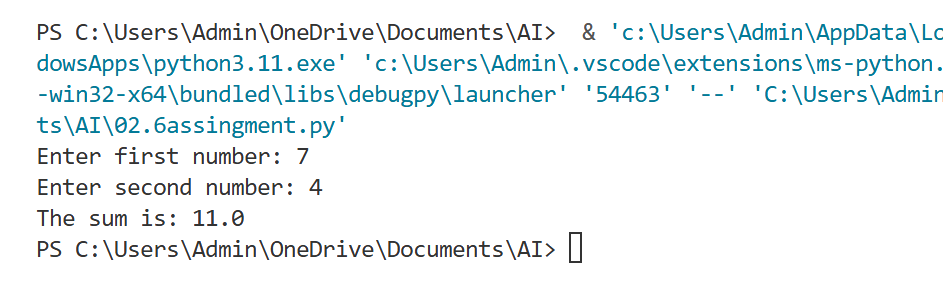


* + **EXPLANATION:**
* **# Define a simple function to add two numbers**
  + This is a comment. It's there for humans to read and understand what the following code does; Python ignores it during execution. It tells us that the code below defines a function for adding two numbers.
* **def add(a, b):**
  + This line starts the definition of a function named add.
  + def is a Python keyword used to define functions.
  + add is the name of the function. You can call the function by this name later.
  + (a, b) are the parameters (or arguments) that the function accepts. When you call this function, you need to provide two values, which will be assigned to the variables a and b inside the function.
  + The colon : at the end of the line signifies the start of the function's code block.
* **return a + b**
  + This line is inside the add function (indicated by the indentation).
  + return is a Python keyword used to send a value back from the function to the place where it was called.
  + a + b calculates the sum of the two parameters a and b that were passed to the function.
  + So, this line calculates the sum and sends that sum back as the result of the function call.
* **# Call the function with some numbers**
  + This is another comment, indicating that the next line will execute the add function.
* **result = add(10, 20)**
  + This line *calls* the add function.
  + add(10, 20) executes the add function with a set to 10 and b set to 20.
  + The value returned by the add function (which is 10 + 20 = 30) is then stored in a variable named result.
* **# Print the result**
  + Another comment, indicating that the next line will display the result.
* **print(f"The result of adding 10 and 20 is: {result}")**
  + This line prints output to the console.
  + print() is a built-in Python function used for output.
  + f"..." denotes an f-string (formatted string literal). This allows you to easily include the value of variables directly within the string.
  + Inside the f-string, The result of adding 10 and 20 is: is plain text that will be printed as is.
  + {result} is a placeholder within the f-string. Python will replace this with the current value of the result variable (which is 30).
  + So, the complete output will be "The result of adding 10 and 20 is: 30".

**VS CODE**



**OUTPUT:**



**EXPLANATION:**

1.Defines a function named [my\_function](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") that takes two arguments ([a](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) and [b](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)) and returns their sum.

2. Prompts the user to enter the first number. The input is converted from a string to a floating-point number and stored in [num1](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

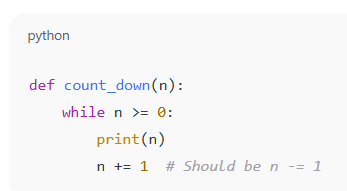
3. Prompts the user to enter the second number. The input is converted to a float and stored in [num2](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).

4. Calls [my\_function](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") with [num1](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) and [num2](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) as arguments, and stores the returned sum in [result](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).

5. Prints the result to the console, displaying the sum of the two numbers.

* **TASK-2**

Identify and fix a logic error in a loop that causes infinite iteration.

****

**Gemini code and output**



**EXPLANATION:**

This code defines a function called count\_down that takes one argument, n, and prints numbers from n down to 0.

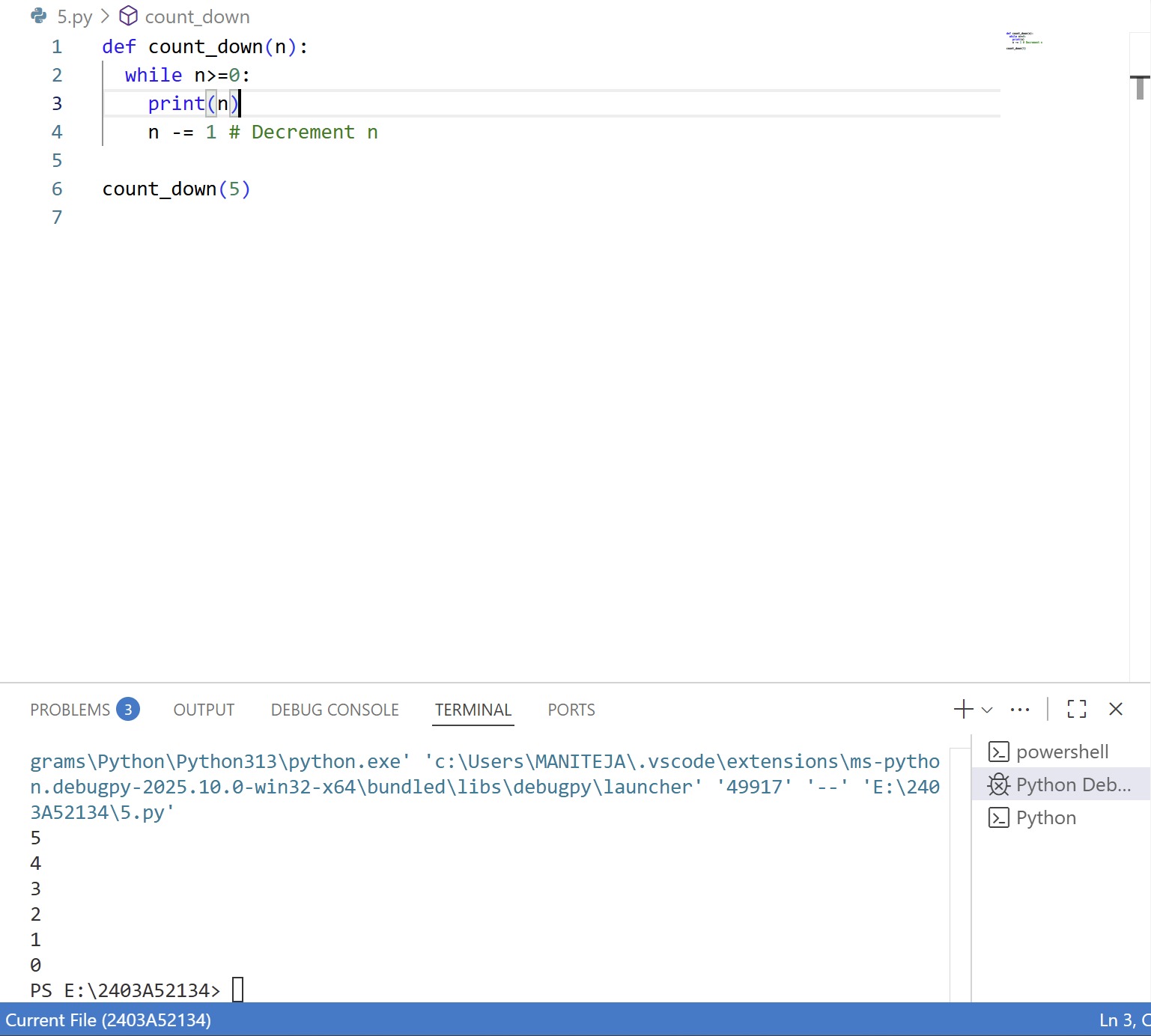
def count\_down(n):  
  while n>=0:  
    print(n)  
    n -= 1 # Decrement n

* def count\_down(n): defines a function named count\_down that accepts one parameter, n.
* while n>=0: starts a while loop that continues as long as the value of n is greater than or equal to 0.
* print(n): Inside the loop, the current value of n is printed.
* n -= 1: After printing, the value of n is decreased by 1. This is the decrement step that moves the count down towards 0.

# Call the function with an initial value for n  
count\_down(5)

* count\_down(5): This line calls the count\_down function with an initial value of 5 for n. The function will then print 5, 4, 3, 2, 1, and 0.

In summary, the count\_down function uses a while loop to repeatedly print the value of n and then decrease n by 1 until n becomes less than 0, at which point the loop terminates.

**VS CODE and output**

**EXPLANATION:**

This code defines a function called count\_down that takes an integer n as input and prints numbers from n down to 0, one per line.

The while n>=0: loop continues as long as n is greater than or equal to 0.

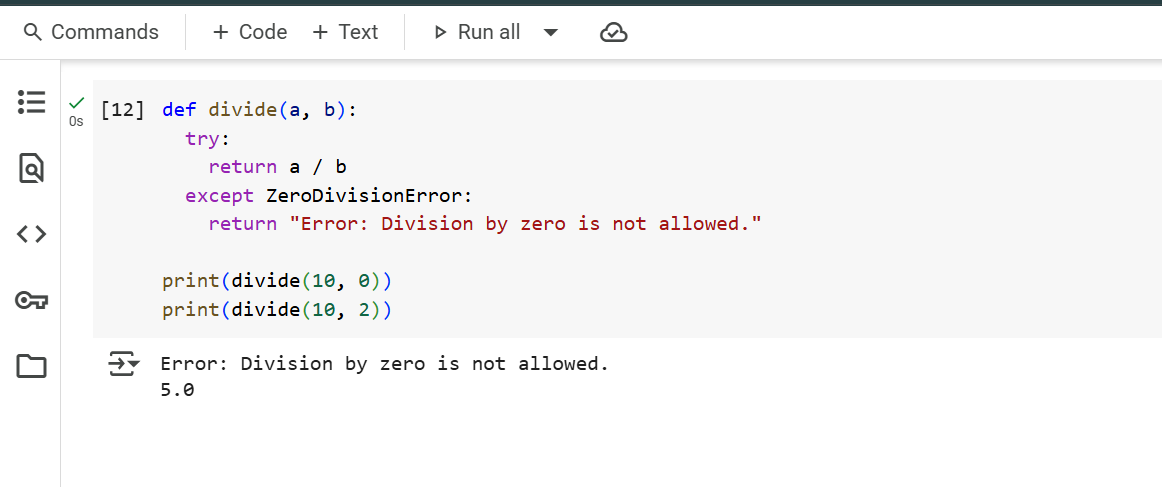
Inside the loop, it prints the current value of n and then decreases n by 1.

After defining the function, count\_down(5) is called, so the output will be:

* **TASK-3**

Debug a runtime error caused by division by zero. Let AI insert try-except.

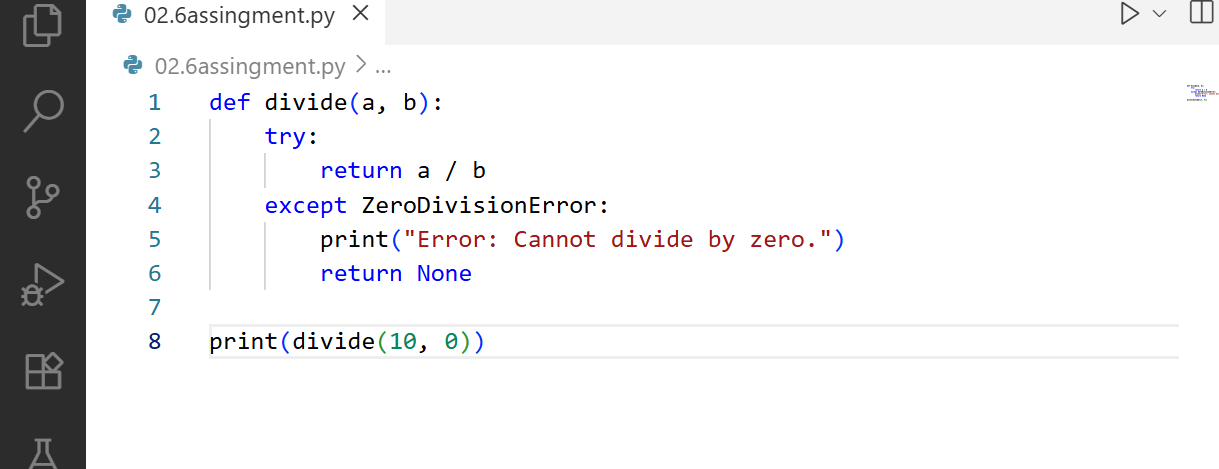
**Gemini code and output**

****

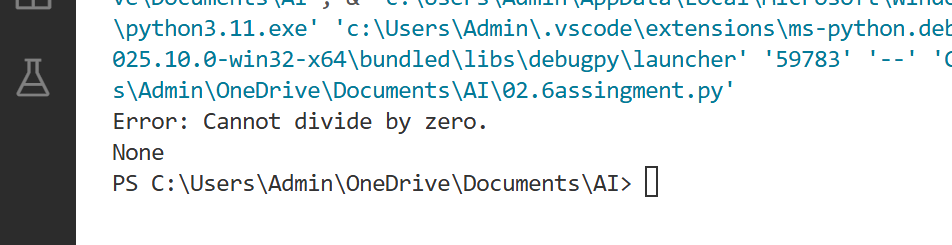
**EXPLANATION:**

* def divide(a, b):
  + This line defines a function named divide that takes two arguments, a and b.
* try:
  + This keyword starts a try block. The code inside this block is attempted to be executed. If an error occurs during the execution of the code within the try block, Python will look for a corresponding except block to handle that specific error.
* return a / b
  + This line is inside the try block. It attempts to perform the division of a by b.
  + If b is not zero, the division will be successful, and the result will be returned by the function.
  + If b is zero, a ZeroDivisionError will occur, and the execution will jump to the except block.
* except ZeroDivisionError:
  + This line starts an except block. It specifies that if a ZeroDivisionError occurs in the preceding try block, the code within this except block should be executed.
* return "Error: Division by zero is not allowed."
  + This line is inside the except block. If a ZeroDivisionError occurred, this line is executed, and the function returns the string "Error: Division by zero is not allowed.".
* print(divide(10, 0))
  + This line calls the divide function with a = 10 and b = 0.
  + Since b is 0, the try block will raise a ZeroDivisionError, and the except block will be executed, returning the error message string.
  + The print() function then displays the returned error message.
* print(divide(10, 2))
  + This line calls the divide function with a = 10 and b = 2.
  + Since b is not zero, the division 10 / 2 is performed successfully within the try block, and the result 5.0 is returned.
  + The print() function then displays the returned value 5.0.

**VS CODE**



**OUTPUT**

****

**EXPLANATION:**

* Defines a function [divide](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) that takes two arguments, [a](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) and [b](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).
* Inside the function, the try block attempts to divide [a](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) by [b](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).
* If [b](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) is zero, a [ZeroDivisionError](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") occurs. The except block catches this error, prints an error message, and returns None.
* Calls the [divide](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) function with arguments 10 and 0.
* Since dividing by zero is not allowed, the error message is printed and None is returned and displayed.
* **TASK-4**

Provide a faulty class definition (missing self in parameters). Let AI fix i

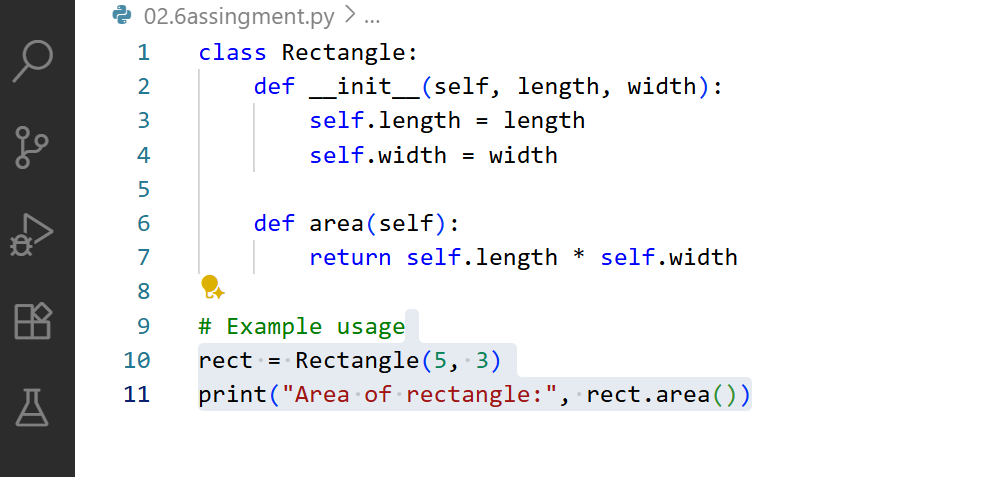
**CODE AND OUTPUT**

**Gemini**

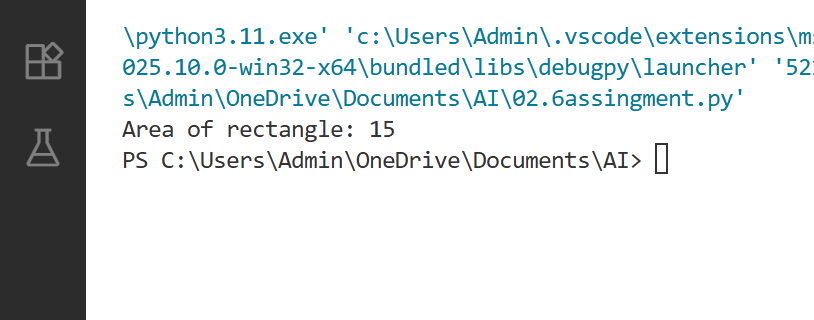
**EXPLANATION:**

* + class Rectangle:
  + This line defines a new class named Rectangle. Classes are blueprints for creating objects (instances).
  + def \_\_init\_\_(self, length, width):
  + This defines the constructor method for the Rectangle class. The \_\_init\_\_ method is called automatically when you create a new Rectangle object. self refers to the instance of the class being created, and length and width are parameters passed when creating an object.
    - self.length = length  
              self.width = width
  + These lines assign the values passed for length and width to attributes of the Rectangle object being created. self.length and self.width become properties of that specific Rectangle instance.
  + def area(self):
  + This defines a method named area within the Rectangle class. This method will calculate and return the area of a Rectangle object. self is used to access the object's attributes.
    - return self.length \* self.width
  + This line calculates the area by multiplying the length and width attributes of the Rectangle object and returns the result.
  + def perimeter(self):
  + This defines another method named perimeter within the Rectangle class. This method will calculate and return the perimeter of a Rectangle object.
    - return 2 \* (self.length + self.width)
  + This line calculates the perimeter using the formula 2 \* (length + width) and returns the result.
  + # Example usage:
  + This is a comment indicating the start of example code that demonstrates how to use the Rectangle class.
  + my\_rectangle = Rectangle(5, 10)
  + This line creates a new instance of the Rectangle class. It calls the \_\_init\_\_ constructor, passing 5 as the length and 10 as the width. The created Rectangle object is assigned to the variable my\_rectangle.
  + print(f"Area: {my\_rectangle.area()}")
  + This line calls the area() method on the my\_rectangle object. The area() method calculates and returns the area, which is then printed to the console along with the text "Area: ".
  + print(f"Perimeter: {my\_rectangle.perimeter()}")
  + This line calls the perimeter() method on the my\_rectangle object. The perimeter() method calculates and returns the perimeter, which is then printed to the console along with the text "Perimeter: ".

**VS CODE**

****

**OUTPUT**

****

**EXPLANATION:**

```python

class Rectangle:

```

Defines a new class named `Rectangle`.

```python

def \_\_init\_\_(self, length, width):

```

Defines the constructor method (`\_\_init\_\_`) that runs when a new `Rectangle` object is created. It takes `length` and `width` as parameters.

```python

self.length = length

```

Assigns the value of `length` to the instance variable `self.length`.

```python

self.width = width

```

Assigns the value of `width` to the instance variable `self.width`.

```python

rect = Rectangle(5,3)

```

Creates a new `Rectangle` object named `rect` with `length` 5 and `width` 3.

```python

print("Length:", rect.length)

```

Prints the string "Length:" followed by the value of `rect.length` (which is 5).

```python

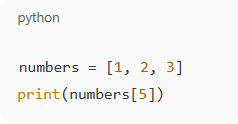
print("Width:", rect.width)

```

Prints the string "Width:" followed by the value of `rect.width` (which is 3).

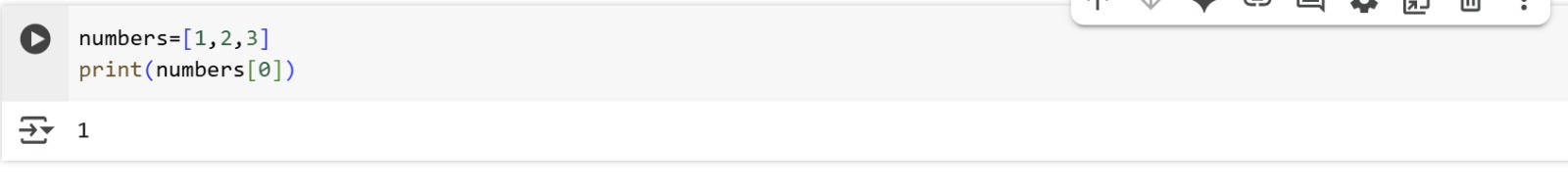
* **TASK-5**

Access an invalid list index and use AI to resolve the Index Error.

****

**CODE AND OUTPUT**

**Gemini**



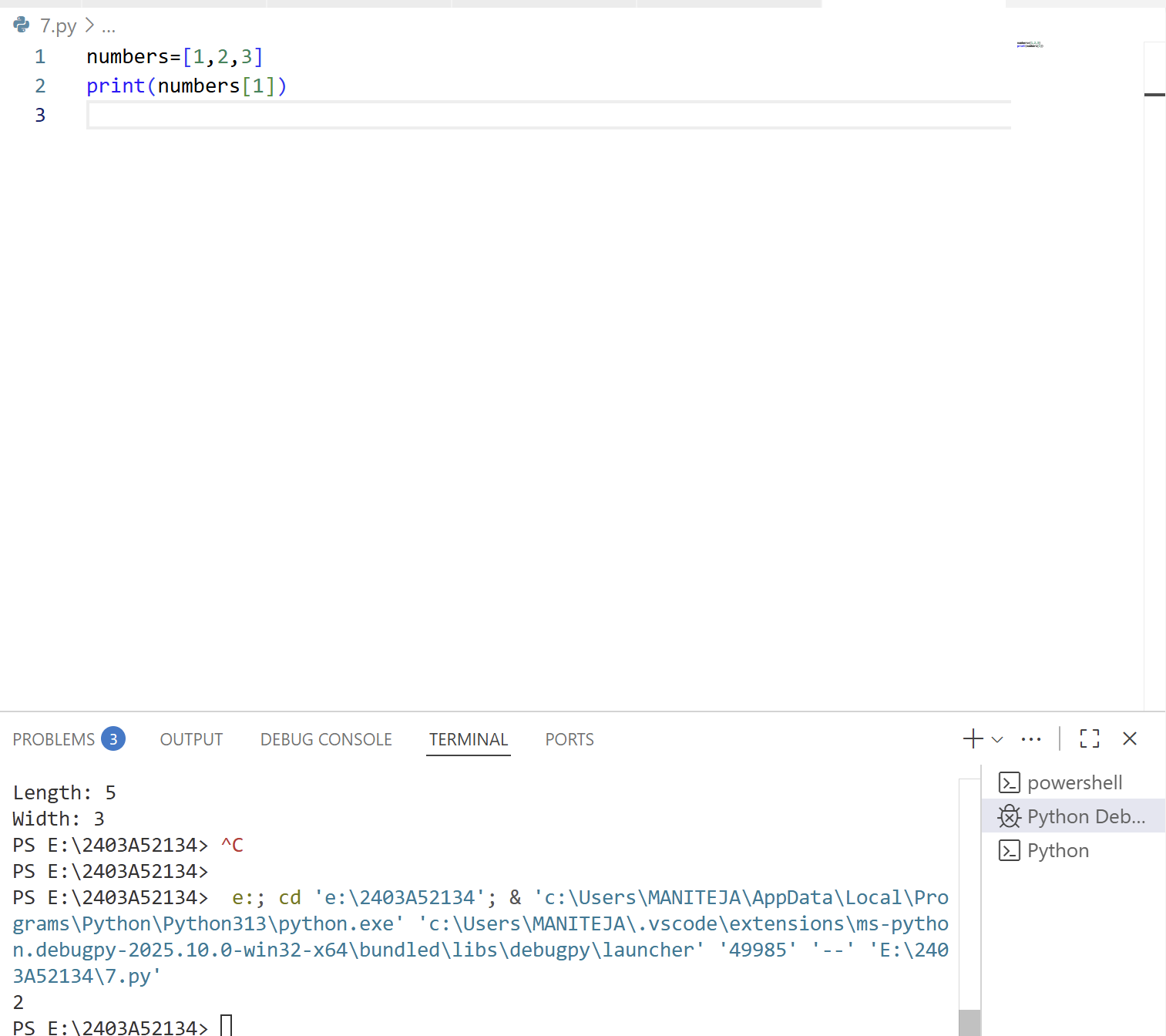
**EXPLANATION:**

numbers=[1,2,3]  
print(numbers[0])

* numbers=[1,2,3]: This line creates a list named numbers and initializes it with three integer elements: 1, 2, and 3. Lists in Python are ordered collections of items, and they are indexed starting from 0. So, the element 1 is at index 0, 2 is at index 1, and 3 is at index 2.
* print(numbers[0]): This line accesses and prints an element from the numbers list.
  + numbers[0] uses square brackets [] with an index (in this case, 0) to access the element at that specific position within the list.
  + Since lists are zero-indexed, numbers[0] refers to the first element in the list, which is the number 1.
  + print() then displays the value of this element to the console.

In summary, this code creates a list of numbers and then prints the first element of that list by accessing it using its index.

**VS CODE AND OUTPUT**



**EXPLANATION:**

This code creates a list called numbers containing three integers: 1, 2, and 3.

It then prints the value at index 1 of the list, which is 2.