ASSIGNMENT – 7.3

2303A51197

Batch-10

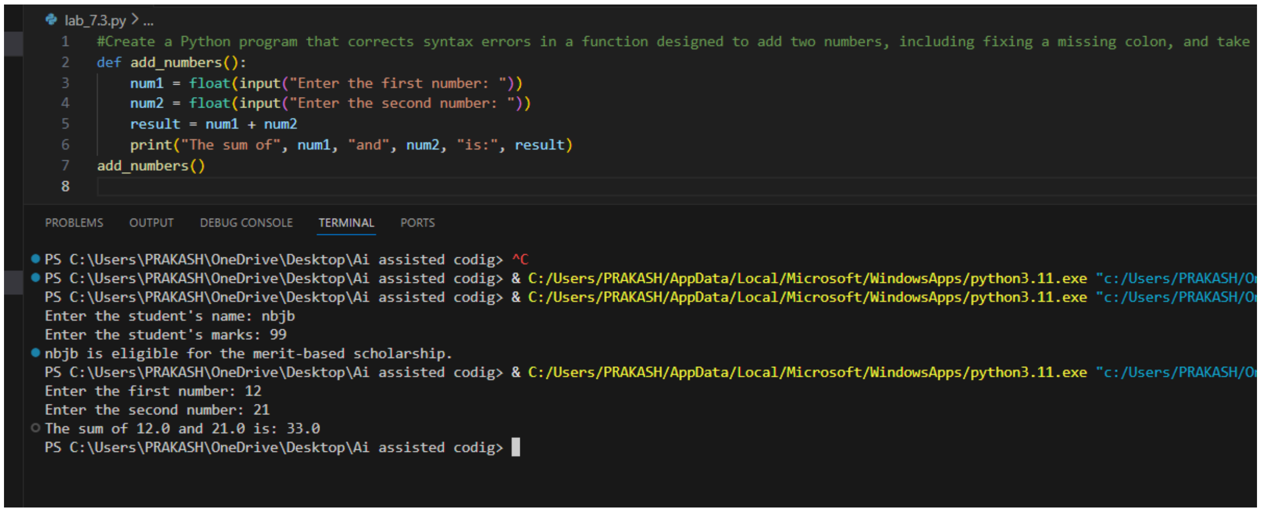
Task-1

Prompt: Create a Python program that corrects syntax errors in a function designed to add two numbers, including fixing a missing colon, and take both numbers as user input.

code :

def add_numbers(num1, num2):   return num1 + num2 # Get user input for the numbers number1 = float(input("Enter the first number: ")) number2 = float(input("Enter the second number: ")) # Call the function and display the result result = add_numbers(number1, number2) print("The sum of", number1, "and", number2, "is:", result)

Output :



```
lab_7.3.py > ...
#Create a Python program that corrects syntax errors in a function designed to add two numbers, including fixing a missing colon, and take
def add_numbers():
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))
    result = num1 + num2
    print("The sum of", num1, "and", num2, "is:", result)
add_numbers()
```

```
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> ^C
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/O
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/O
Enter the student's name: nbjb
Enter the student's marks: 99
nbjb is eligible for the merit-based scholarship.
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/O
Enter the first number: 12
Enter the second number: 21
The sum of 12.0 and 21.0 is: 33.0
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig>
```

Code Analysis:

- The function add_numbers() takes two parameters and returns their sum.
- The missing colon after the function definition is corrected.
- User inputs are converted to float to allow decimal values.
- The function is called with user inputs and the result is printed.
- Using functions improves reusability and modular programming.

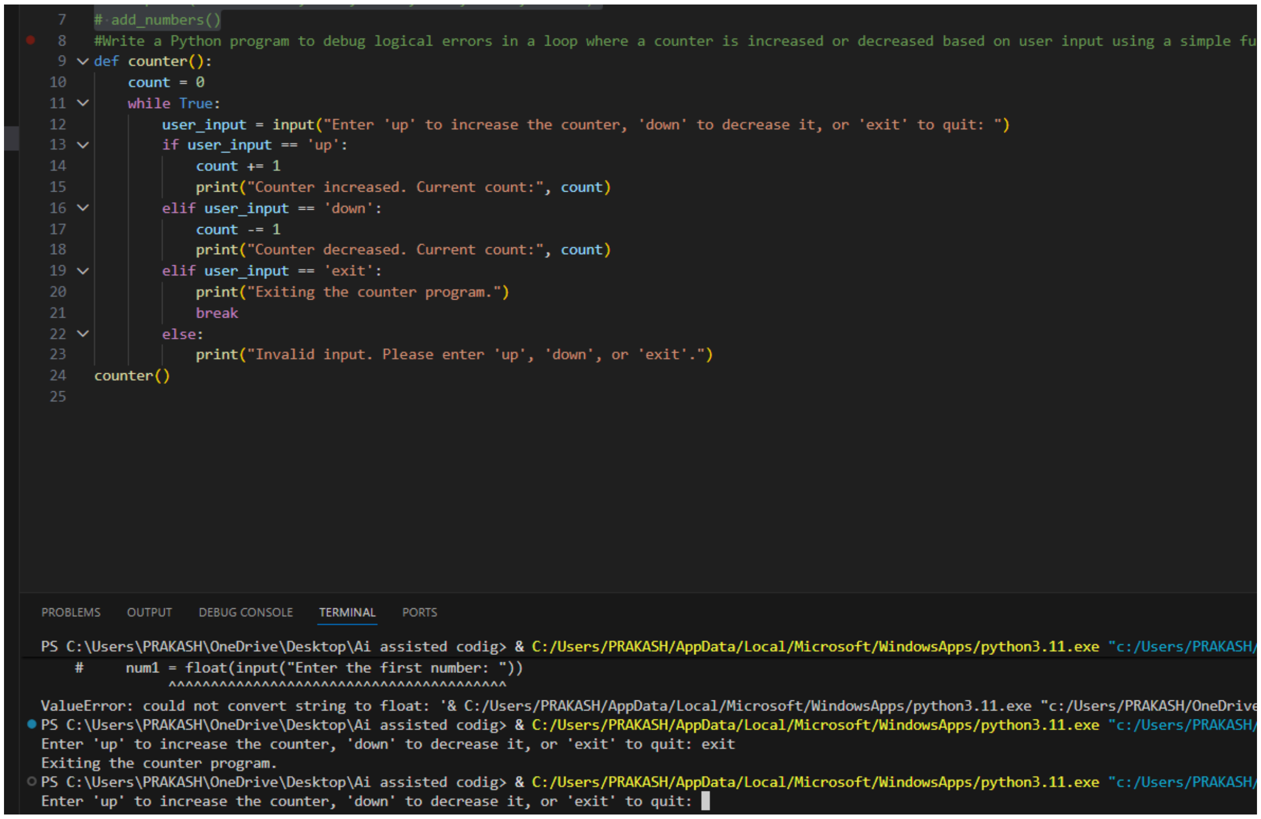
Task-2

Prompt: Write a Python program to debug logical errors in a loop where a counter is increased or decreased based on user input using a simple function.

Code :

```
def update_counter(counter, action):

if action == 'increment':         return

counter + 1      elif action == 'decrement':

        return counter - 1

else:

        return counter #

Initialize counter counter

= 0

# Taking user input for action action = input("Enter

action (increment/decrement): ")

# Updating counter based on user input and printing

the result counter = update_counter(counter, action)

print(f"Counter value after {action}: {counter}")
```

Output :



```
# add_numbers()
#Write a Python program to debug logical errors in a loop where a counter is increased or decreased based on user input using a simple fu
def counter():
    count = 0
    while True:
        user_input = input("Enter 'up' to increase the counter, 'down' to decrease it, or 'exit' to quit: ")
        if user_input == 'up':
            count += 1
            print("Counter increased. Current count:", count)
        elif user_input == 'down':
            count -= 1
            print("Counter decreased. Current count:", count)
        elif user_input == 'exit':
            print("Exiting the counter program.")
            break
        else:
            print("Invalid input. Please enter 'up', 'down', or 'exit'.")
counter()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/
    #     num1 = float(input("Enter the first number: "))
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

ValueError: could not convert string to float: '& C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/
Enter 'up' to increase the counter, 'down' to decrease it, or 'exit' to quit: exit
Exiting the counter program.
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/
Enter 'up' to increase the counter, 'down' to decrease it, or 'exit' to quit: █
```

Code Analysis:

- The function modifies the counter based on user action.
- action.lower() avoids case-sensitivity issues.
- If invalid input is entered, the counter remains unchanged.
- The logic ensures proper increment/decrement functionality.
- This demonstrates basic debugging of logical conditions.

Task-3

Prompt: Develop a Python function that handles runtime errors such as division by zero using try and except blocks without prior validation, and accept input values from the user

Code : def

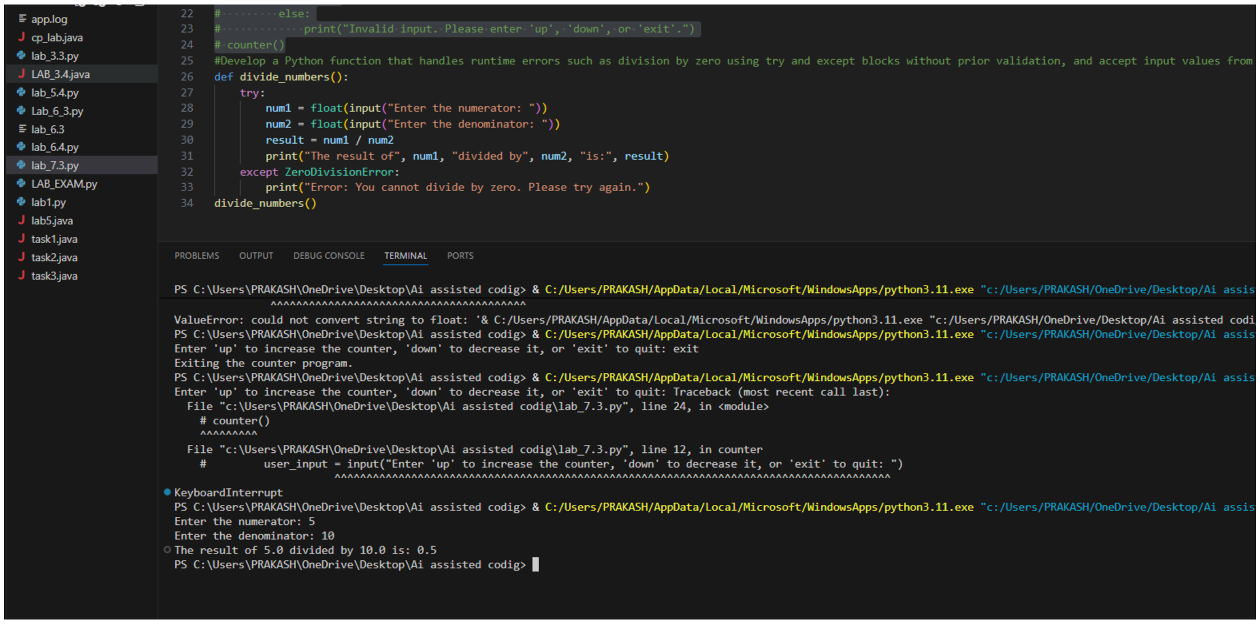
divide_numbers(num1, num2):

try:

result = num1 / num2 return result

except ZeroDivisionError: return "Error: Division by zero is not allowed." # Get user input for the numbers number1 = float(input("Enter the numerator: ")) number2 = float(input("Enter the denominator: ")) # Call the function and display the result result = divide_numbers(number1, number2) print("The result of dividing", number1, "by", number2, "is:", result) Output:

Code Analysis :



```python
#·········else:
#·············print("Invalid input. Please enter 'up', 'down', or 'exit'.")
# counter()
#Develop a Python function that handles runtime errors such as division by zero using try and except blocks without prior validation, and accept input values from
def divide_numbers():
    try:
        num1 = float(input("Enter the numerator: "))
        num2 = float(input("Enter the denominator: "))
        result = num1 / num2
        print("The result of", num1, "divided by", num2, "is:", result)
    except ZeroDivisionError:
        print("Error: You cannot divide by zero. Please try again.")
divide_numbers()
```

```
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai assis
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

ValueError: could not convert string to float: '& C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai assisted codi
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai assis
Enter 'up' to increase the counter, 'down' to decrease it, or 'exit' to quit: exit
Exiting the counter program.
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai assis
Enter 'up' to increase the counter, 'down' to decrease it, or 'exit' to quit: Traceback (most recent call last):
  File "c:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig\lab_7.3.py", line 24, in <module>
    # counter()
    ^^^^^^^^^
  File "c:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig\lab_7.3.py", line 12, in counter
    #         user_input = input("Enter 'up' to increase the counter, 'down' to decrease it, or 'exit' to quit: ")
                           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai assis
Enter the numerator: 5
Enter the denominator: 10
The result of 5.0 divided by 10.0 is: 0.5
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> █
```

- The function attempts division inside a try block.
- If the denominator is zero, ZeroDivisionError is caught.
- The program does not crash due to exception handling.
- A user-friendly error message is returned instead.
- try-except ensures runtime stability.

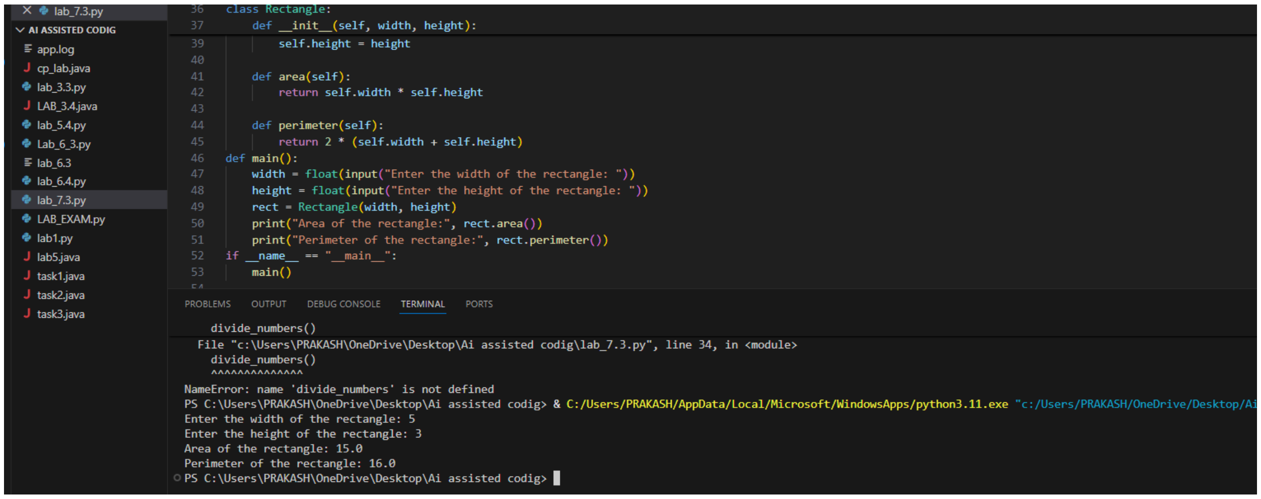
Task-4

Prompt: #generate a code to debug the class definition errors for a rectangle .provide a class definition with missing self-parameter and correct it using inti method and explain why self is used in class definitions .take user input

Code :

```
class Rectangle:     def
__init__(self, width, height):
        self.width = width          self.height = height      def
area(self):           return self.width * self.height # Get user
input for width and height width = float(input("Enter the
width of the rectangle: ")) height = float(input("Enter the
height of the rectangle: ")) # Create an instance of the
Rectangle class rectangle = Rectangle(width, height) #
Calculate and display the area of the rectangle print("The
area of the rectangle is:", rectangle.area())
# Explanation: The self parameter is used in class definitions #
to refer to the instance of the class. It allows us to access and
modify the attributes of the instance.
# In the __init__ method, we use self to assign the width and height
values to the instance variables.
```

Output:



```python
class Rectangle:
    def __init__(self, width, height):
        self.height = height

    def area(self):
        return self.width * self.height

    def perimeter(self):
        return 2 * (self.width + self.height)
def main():
    width = float(input("Enter the width of the rectangle: "))
    height = float(input("Enter the height of the rectangle: "))
    rect = Rectangle(width, height)
    print("Area of the rectangle:", rect.area())
    print("Perimeter of the rectangle:", rect.perimeter())
if __name__ == "__main__":
    main()
```

```
    divide_numbers()
  File "c:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig\lab_7.3.py", line 34, in <module>
    divide_numbers()
    ^^^^^^^^^^^^^^
NameError: name 'divide_numbers' is not defined
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai
Enter the width of the rectangle: 5
Enter the height of the rectangle: 3
Area of the rectangle: 15.0
Perimeter of the rectangle: 16.0
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> 
```

Code Analysis :

- The constructor method must be __init__ (double underscores).
- self refers to the current object instance.
- Instance variables (self.width, self.height) store object data.
- The area() method accesses instance variables using self.
- Without self, Python cannot link data to the specific object.

Task-5

Prompt: Write a Python program that demonstrates and fixes index errors in lists by handling out-of-range access using exception handling, and explain why managing index errors is important, while taking list elements from user input.

my_list = [1, 2, 3] try:

    # Attempting to access an out-of-range index

print(my_list[5]) except IndexError:

    print("Error: Index out of range. Please provide a valid index.")

# Get user input for list elements user_input = input("Enter a list of

numbers separated by commas: ")

# Convert the user input into a list of integers my_list

= [int(x.strip()) for x in user_input.split(",")] # Attempt to access

an index based on user input try: index = int(input("Enter the

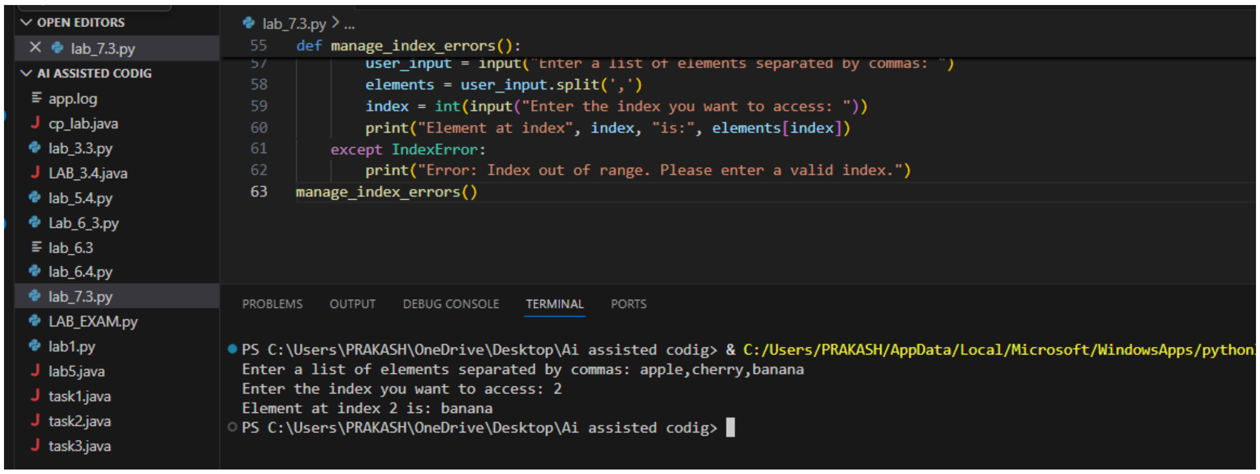
index you want to access: "))   print("Element at index", index,

"is:", my_list[index]) except IndexError: print("Error: Index

out of range. Please provide a valid index.")

# Explanation: Handling index errors in list operations is important because it prevents the program from crashing when an invalid index is accessed. By using exception handling, we can catch the error and provide a user-friendly message, allowing the program to continue running smoothly even when unexpected input is encountered.

Output :



```python
def manage_index_errors():
        user_input = input("Enter a list of elements separated by commas: ")
        elements = user_input.split(',')
        index = int(input("Enter the index you want to access: "))
        print("Element at index", index, "is:", elements[index])
    except IndexError:
        print("Error: Index out of range. Please enter a valid index.")
manage_index_errors()
```

```
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:/Users/PRAKASH/AppData/Local/Microsoft/WindowsApps/python
Enter a list of elements separated by commas: apple,cherry,banana
Enter the index you want to access: 2
Element at index 2 is: banana
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig>
```

Code Analysis :

- User input is converted into a list using split() and list comprehension.
- The program attempts to access a user-specified index.
- If index is invalid, IndexError is handled gracefully.
- ValueError ensures proper numeric input.
- Exception handling prevents program crashes and improves reliability