

ASSIGNMENT

NAME: B. Akhira Nandhini

HTNO:2303A51516

BATCH:22

AI ASSISTANT CODING

Task Description-1

- Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime

Expected Output-1

- A basic Python function to check if a number is prime, demonstrating correct logical conditions without relying on examples or additional context

INPUT and OUTPUT:

```
File Edit Selection View Go Run Terminal Help ← → Q, ai assisted coding
EXPLORER Welcome task1.py X
AI ASSISTED CODING assignmet 4.2
task1.py
Ai assistant 1.docx
Ai assistant 2.docx
ai assisted coding 1.docx
ai assisted coding 2.docx
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assited coding> & c:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assited coding/assignmet 4.2/task1.py"
Enter a number: 9
9 is not a prime number.
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assited coding>
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assited coding>
```

The screenshot shows a code editor interface with a dark theme. The left sidebar has a tree view labeled 'EXPLORER' showing files like 'task1.py' and various 'docx' files. The main area shows a Python script named 'task1.py'. The code defines a function 'is_prime(n)' that checks if a number is prime by iterating from 2 to the square root of n. The script then prompts the user for input, calls the function, and prints the result. Below the code editor is a terminal window showing the command to run the script and its output: '9 is not a prime number.'. The bottom status bar shows system information like battery level (28°C), date (20-01-2026), and time (14:00).

Explanation:

- A number that is less than or equal to 1 cannot be a prime number, so first we check for that condition.

- Then, to find whether the number is prime or not, we divide it by numbers starting from 2 up to the square root of the given number.
- If the number gets divided by any of these values, it means it has factors and is not a prime number.
- If no divisor is found in this range, then the number has only two factors (1 and itself), so it is a prime number.

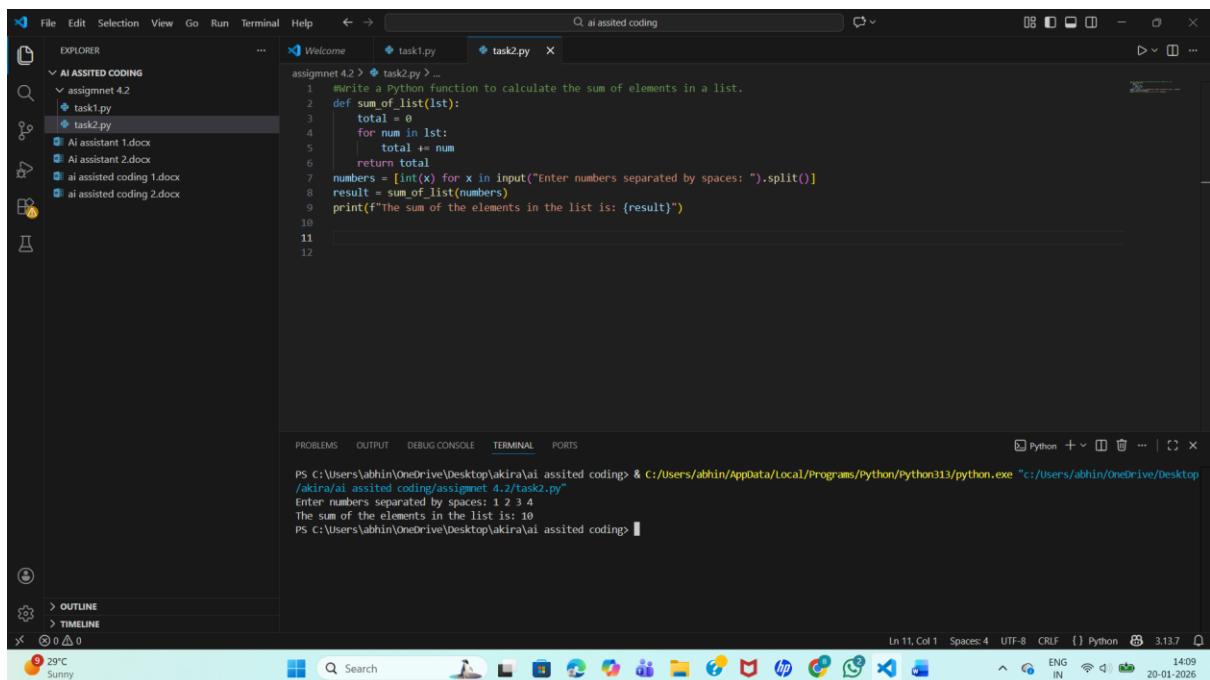
Task Description-2

- One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.

Expected Output-2

- A correct conversion function guided by the single example.

INPUT and OUTPUT:



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar (EXPLORER) lists files: 'AI ASSISTED CODING', 'assigmnent-4.2', 'task1.py', 'task2.py', 'Ai assistant 1.docx', 'Ai assistant 2.docx', 'ai assisted coding 1.docx', and 'ai assisted coding 2.docx'. The center area (EDITOR) contains a Python script named 'task2.py' with the following code:

```

assignmnent 4.2 > task2.py ...
1 #write a Python function to calculate the sum of elements in a list.
2 def sum_of_list(lst):
3     total = 0
4     for num in lst:
5         total += num
6     return total
7 numbers = [int(x) for x in input("Enter numbers separated by spaces: ").split()]
8 result = sum_of_list(numbers)
9 print(f"The sum of the elements in the list is: {result}")
10
11
12

```

The bottom right terminal window shows the output of running the script:

```

PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & c:/Users/abhin/AppData/Local/Programs/Python/python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assigmnent 4.2/task2.py"
Enter numbers separated by spaces: 1 2 3 4
The sum of the elements in the list is: 10
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>

```

The status bar at the bottom indicates: Ln 11, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | 3.13.7 | ENG IN | 14:09 | 20-01-2026.

EXPLANATION:

- In this task, the function is used to find the sum of all elements present in a list.
- First, a variable is initialized to store the total sum.
- Then, the function goes through each element in the list one by one using a loop and adds each value to the total.

- After all the elements are added, the final sum is returned as the output.
- This approach is simple and clearly shows how the sum of a list can be calculated using basic logic without directly using any built-in function.

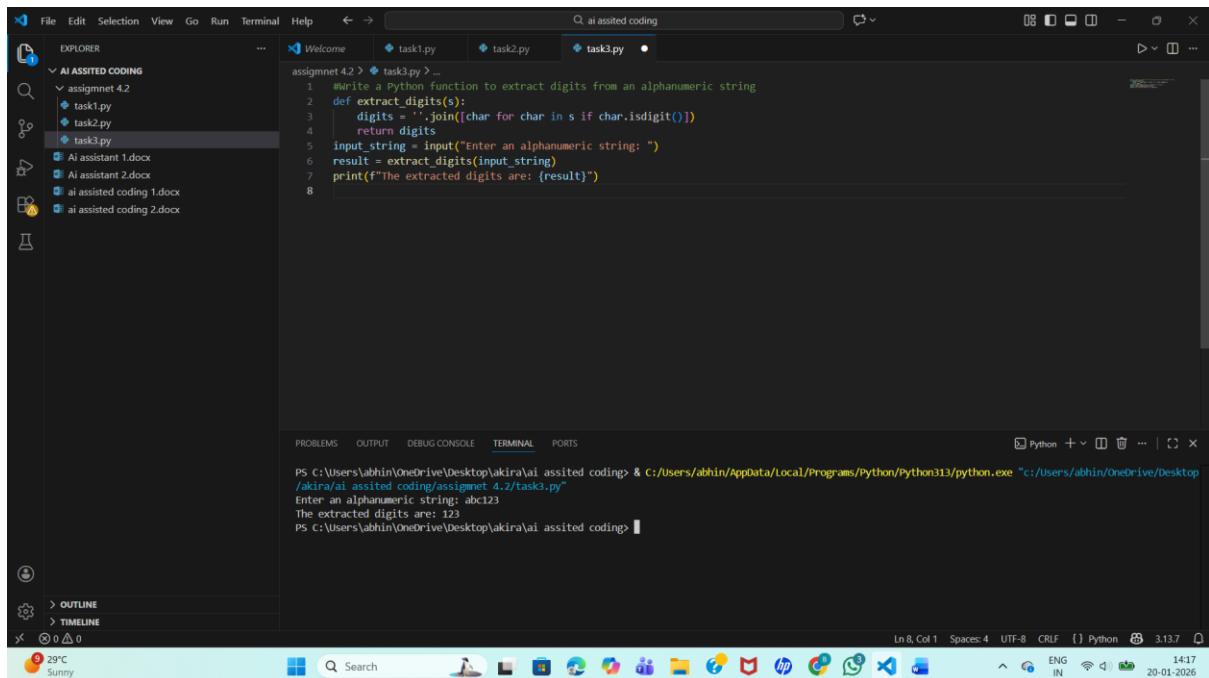
Task Description-3

- Few-shot: Give 2–3 examples to create a function that extracts digits from an alphanumeric string.

Expected Output-3

- Accurate function that returns only the digits from alphanumeric string.

INPUT AND OUTPUT:



```
assignmet 4.2 > task3.py >-
1 #write a Python function to extract digits from an alphanumeric string
2 def extract_digits(s):
3     digits = ''.join([char for char in s if char.isdigit()])
4     return digits
5 input_string = input("Enter an alphanumeric string: ")
6 result = extract_digits(input_string)
7 print(f"The extracted digits are: {result}")
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\abhin\OneDrive\Desktop\akira\ai assited coding> & c:/Users/abhin/AppData/Local/Programs/Python/python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assited coding/assignmet 4.2/task3.py"
Enter an alphanumeric string: abc123
The extracted digits are: 123
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assited coding>

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.7

29°C Sunny

EXPLANATION:

- In this task, the function is designed to extract only the digits from an alphanumeric string.
- The function checks each character of the given string one by one.
- If the character is a number, it is added to a new string, and if it is a letter or any other symbol, it is ignored.
- By the end of the loop, the new string contains only the digits from the original input.

- This method makes it easy to separate numbers from text and works correctly for any alphanumeric string.

Task Description-4

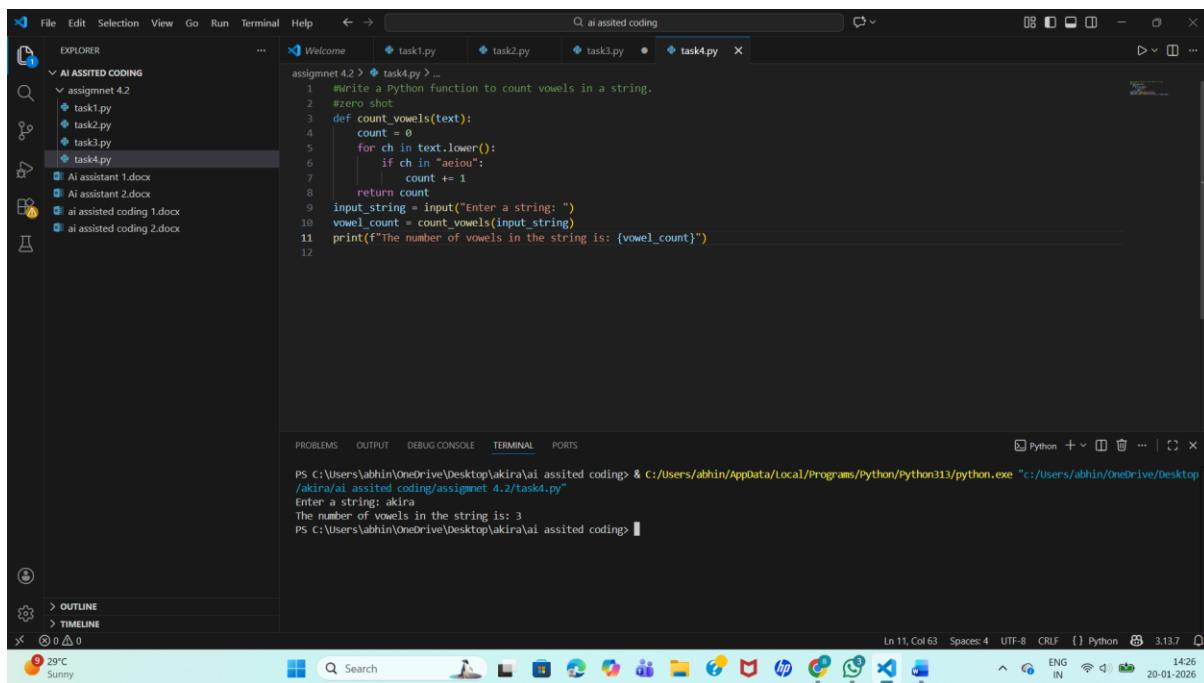
- Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.

Expected Output-4

- Output comparison + student explanation on how examples helped the model.

INPUT AND OUTPUT:

Zero shot:



The screenshot shows a dark-themed instance of Visual Studio Code. In the center, a code editor window displays a Python file named 'task4.py'. The code defines a function 'count_vowels' that takes a string 'text' as input, iterates through its lowercase characters, and increments a counter if the character is one of 'aeiou'. It then prints the total count. Below the code editor, a terminal window shows the command 'python task4.py' being run, followed by the output 'The number of vowels in the string is: 3'. The left sidebar features a 'File Explorer' panel with a tree view of files under 'AI ASSISTED CODING' and a 'Terminal' tab at the bottom. The bottom right corner shows system status icons like battery level, signal strength, and date/time.

```

1 #Write a Python function to count vowels in a string.
2 #zero shot
3 def count_vowels(text):
4     count = 0
5     for ch in text.lower():
6         if ch in "aeiou":
7             count += 1
8     return count
9 input_string = input("Enter a string: ")
10 vowel_count = count_vowels(input_string)
11 print(f"The number of vowels in the string is: {vowel_count}")
12

```

Few shot:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files: assignmet 4.2, task1.py, task2.py, task3.py, task4.py, Ai assistant 1.docx, Ai assistant 2.docx, ai assisted coding 1.docx, and ai assisted coding 2.docx. The main editor window displays Python code for counting vowels in a string. The terminal below shows two runs of the code, one with input 'akira' and another with 'hello', both outputting the result 'The number of vowels in the string is: 3'. The status bar at the bottom indicates the file is 4.2/task4.py, the line is 19, column 63, and the date is 20-01-2026.

```

12
13 #few shot
14 def count_vowels(text):
15     vowels = "aeiou"
16     return sum(1 for ch in text.lower() if ch in vowels)
17 input_string = input("Enter a string: ")
18 vowel_count = count_vowels(input_string)
19 print(f"The number of vowels in the string is: {vowel_count}")
20

```

EXPLANATION:

- In this task, zero-shot and few-shot prompting are compared using a function that counts the number of vowels in a string.
- In the zero-shot approach, only the instruction is given, so the function is written using basic logic with loops and conditions.
- In the few-shot approach, examples are provided, which help in understanding the pattern more clearly and allow the function to be written in a more optimized and concise way.
- By comparing both outputs, it is observed that while both methods give correct results, few-shot prompting improves clarity, efficiency, and overall quality of the generated code because the examples guide the model better.

Task Description-5

- Use few-shot prompting with 3 sample inputs to generate a function that determines

the minimum of three numbers without using the built-in min() function.

Expected Output-5

- A function that handles all cases with correct logic based on example patterns.

INPUT AND OUTPUT:

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a folder named "assigmnent 4.2" containing files: task1.py, task2.py, task3.py, task4.py, and task5.py. task5.py is currently selected.
- Code Editor:** Displays the following Python code:

```
1 # TASK5
2 # Write a Python function to find the minimum of three numbers without using the built-in min() function.
3 # Examples:
4 # Input: (3, 7, 5) -> Output: 3
5 # Input: (10, 2, 8) -> Output: 2
6 # Input: (-1, 4, 0) -> Output: -1
7
8 def find_minimum(a, b, c):
9     if a <= b and a <= c:
10         return a
11     elif b <= a and b <= c:
12         return b
13     else:
14         return c
15
16 # Example usage
17 num1, num2, num3 = 3, 7, 5
18 minimum = find_minimum(num1, num2, num3)
19 print(f"The minimum of {num1}, {num2}, {num3} is {minimum}.")
```
- Terminal:** Shows the command line output:

```
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assigmnent 4.2/task5.py"
The minimum of 3, 7, 5 is 3.
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>
```
- Bottom Status Bar:** Shows system information including weather (29°C, Sunny), date (20-01-2026), and time (14:34).

EXPLANATION:

- In this task, the function is written to find the minimum of three numbers without using the built-in `min()` function.
- The function compares the three values using conditional statements. First, it checks whether the first number is smaller than or equal to the other two.
- If not, it then checks the second number in the same way. If neither of these conditions is true, the third number is considered the smallest.
- This logic ensures that all possible cases are handled correctly, and the correct minimum value is returned based on the given examples.