

**PROGRAM** : B.TECH/CSE  
**SPECIALIZATION** : AIML  
**COURSE TITLE** : AI ASSISTANT CODING  
**COURSE CODE** : 24CS101PC214  
**SEMESTER** : 3RD  
**NAME OF THE STUDENT** : KARNAKANTI GODADEVI  
**ENROLLMENT NO** : 2403A52003  
**BATCH NO** : 01

**Task Description#1**

- Zero-shot: Prompt AI with only the instruction — Write a Python function to generate the Fibonacci sequence up to n terms.

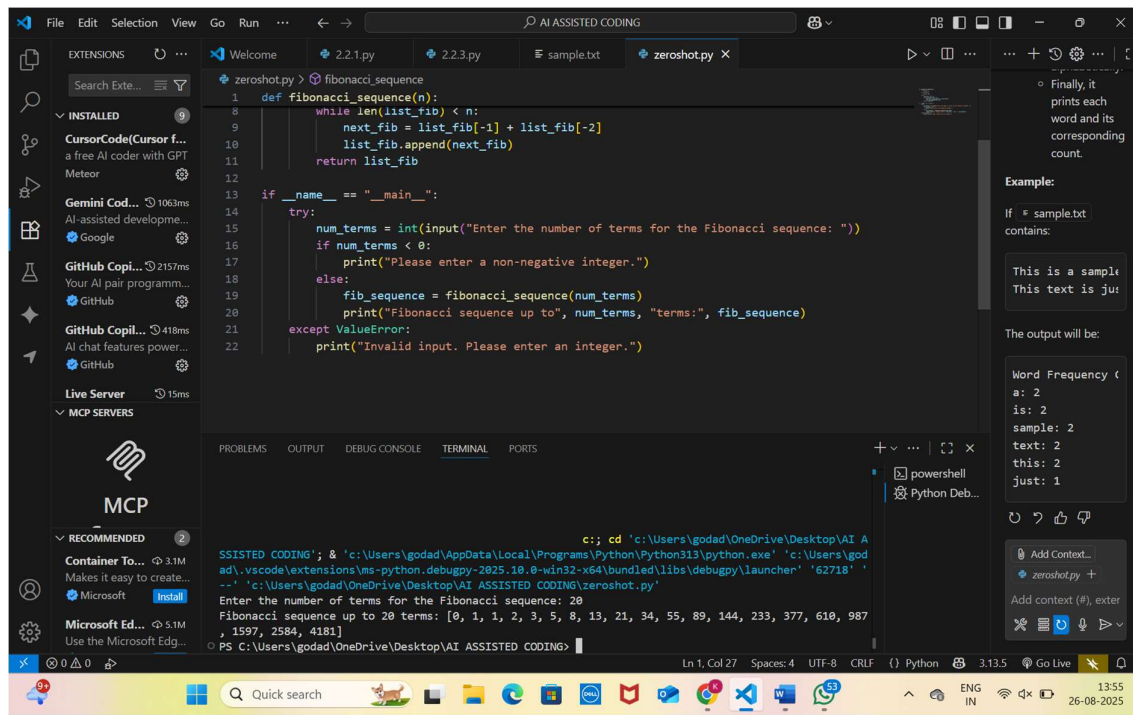
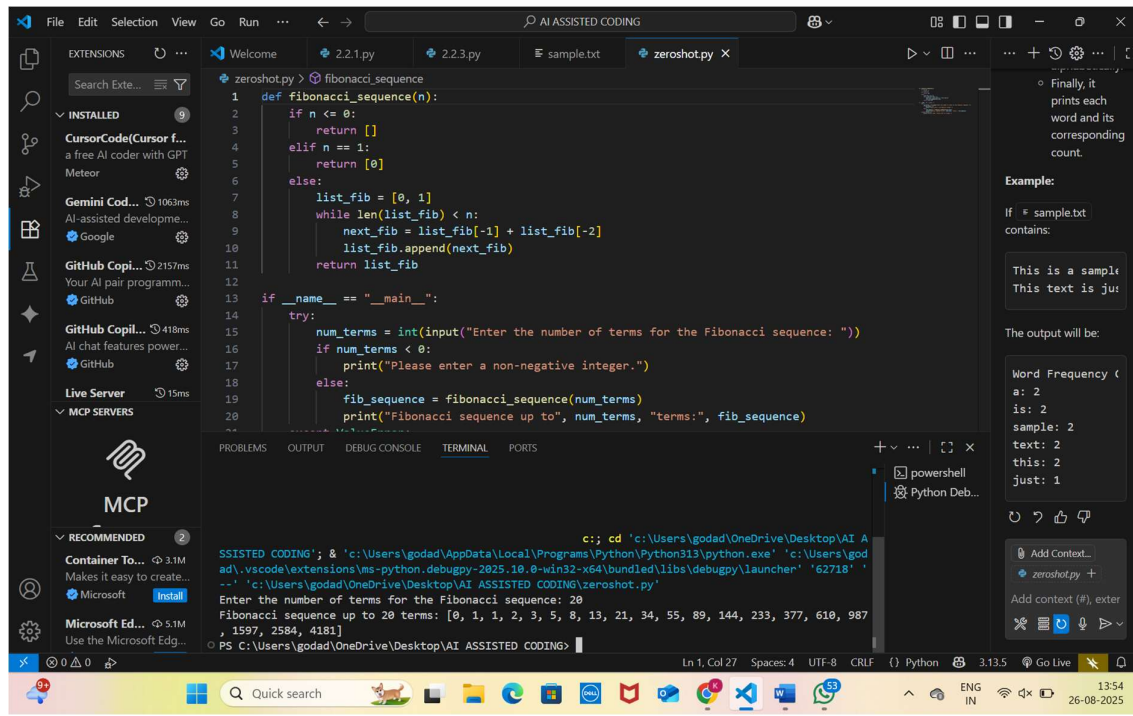
**Expected Output#1**

- A working function without using any sample inputs/outputs.

**Prompt:**

Write a Python function to generate the Fibonacci sequence up to n terms dynamically generate output by asking the user the number of termss.

**Code & Output:**



## Task Description#2

- One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius.

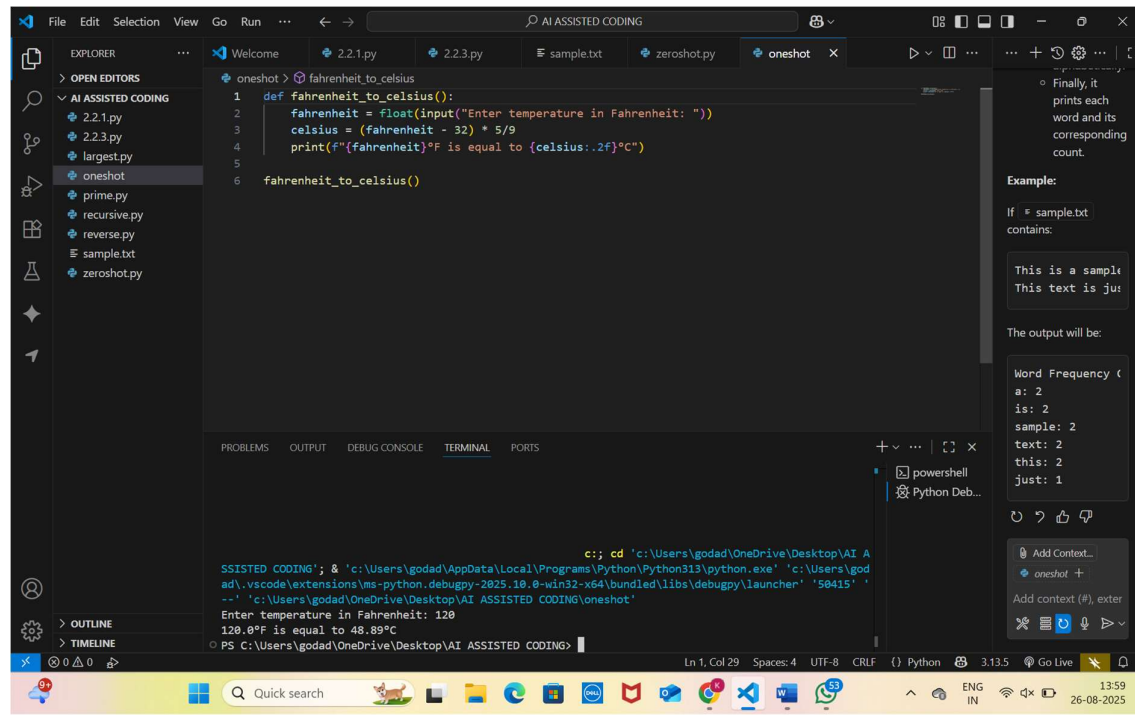
## Expected Output#2

- A correct conversion function guided by the single example.

### Prompt:

example: Input: 100, Output: 37.78 generate a function that converts Fahrenheit to Celsius by asking the user dynamically and generate output.

### Code&Output:



### Task Description#3

- Few-shot: Give 2–3 examples to create a function that extracts the domain name from an email address.

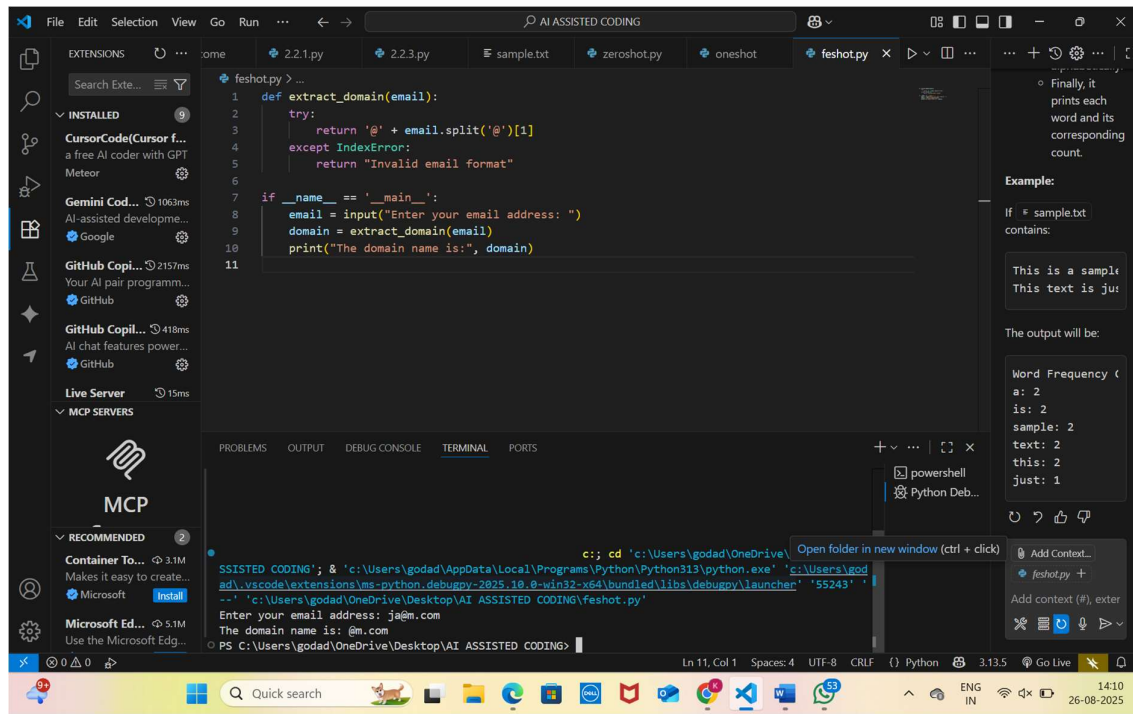
### Expected Output#3

- Accurate function that returns only the domain portion of an email (e.g., @gmail.com).

### Prompt:

Examples: [1.2403a52003@sru.edu.in](mailto:1.2403a52003@sru.edu.in) and domain is @sru.edu.in  
[2.godadevireddy@gmail.com](mailto:2.godadevireddy@gmail.com) and domain is @gmail.com. By these examples create a function that extracts the domain name from an email address by asking the user email address dynamically.

### Code&Output:



#### Task Description#4

- Compare zero-shot vs few-shot prompting for generating a function that checks whether a word is a palindrome, ignoring punctuation and case.

#### Expected Output#4

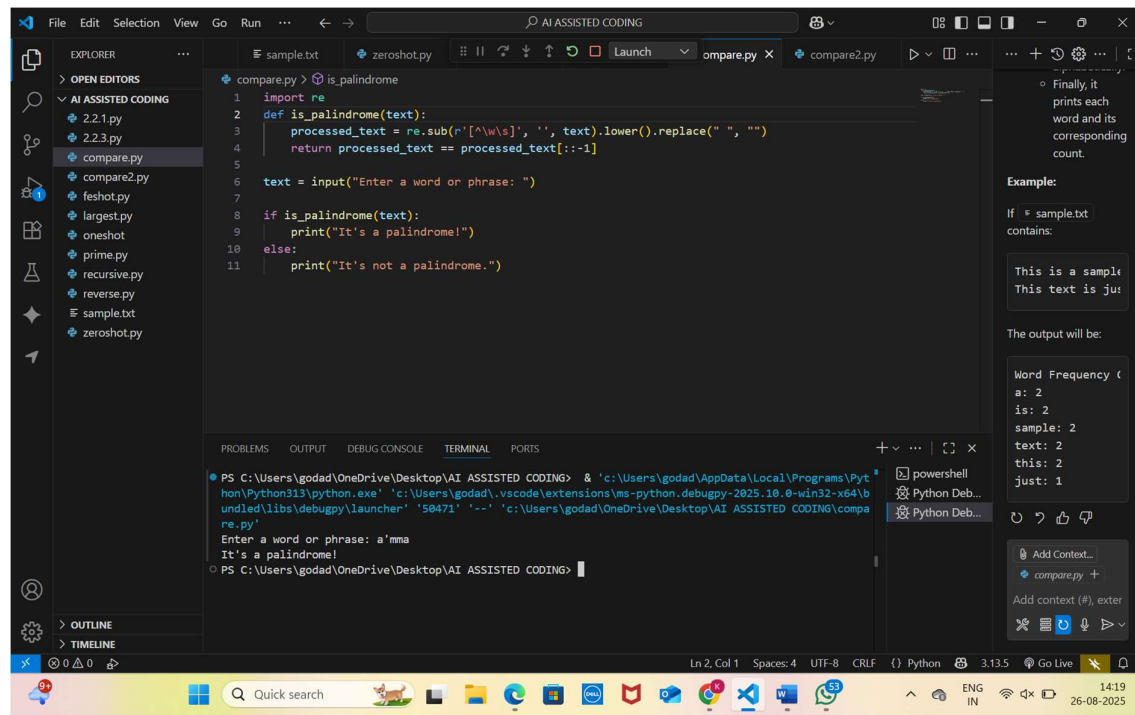
- Output comparison + student explanation on how examples helped the mode

#### Prompt:

#### Zeroshot:

\_check whether the given word is palindrome or not ignoring punctuation and case by asking user dynamically.

## Code&Output:



The screenshot shows a VS Code editor with a Python script named `compare.py` open. The script defines a function `is_palindrome` that takes a string `text` and returns a boolean. It uses `re.sub` to remove non-alphanumeric characters and `lower()` to convert the string to lowercase. The script then prompts the user to enter a word or phrase and prints the result.

```
1 import re
2 def is_palindrome(text):
3     processed_text = re.sub(r'[^a-zA-Z]', '', text).lower().replace(" ", "")
4     return processed_text == processed_text[::-1]
5
6 text = input("Enter a word or phrase: ")
7
8 if is_palindrome(text):
9     print("It's a palindrome!")
10 else:
11     print("It's not a palindrome.")
```

The terminal output shows the script being executed. The user enters `a'mma`, and the output is `It's a palindrome!`.

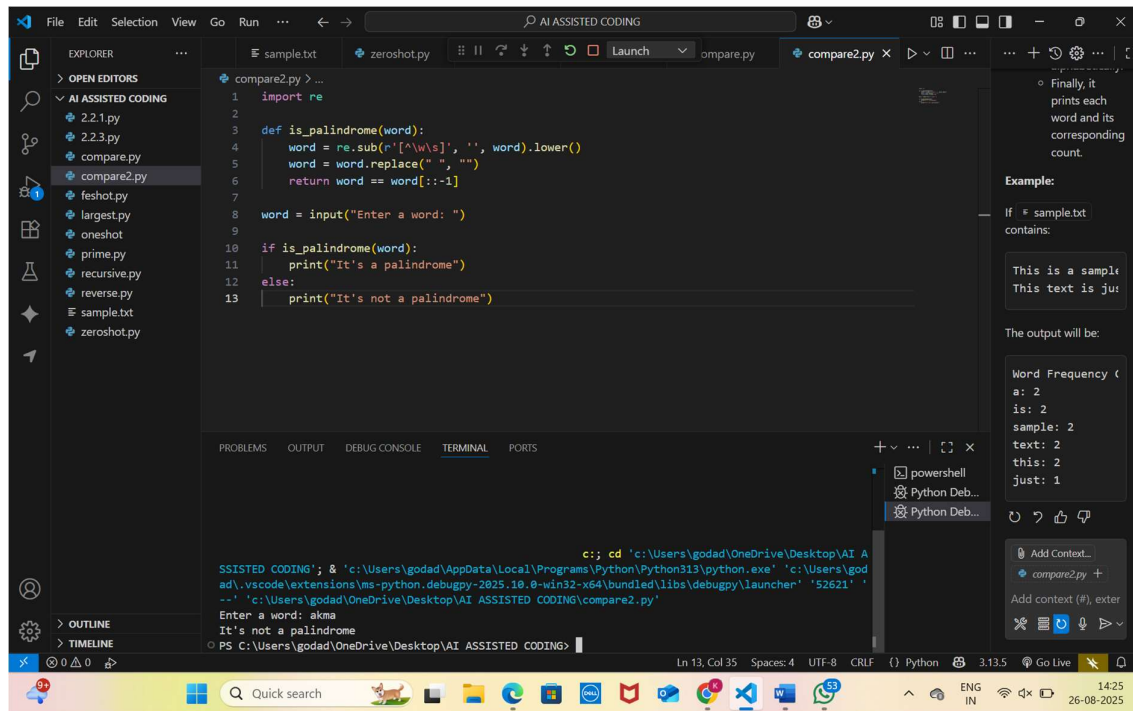
```
PS C:\Users\godad\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\godad\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\godad\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bin\debugpy\launcher' '58471' '--' 'c:\Users\godad\OneDrive\Desktop\AI ASSISTED CODING\compare.py'
Enter a word or phrase: a'mma
It's a palindrome!
PS C:\Users\godad\OneDrive\Desktop\AI ASSISTED CODING>
```

On the right side of the editor, there is a panel showing the output of the script. It includes a note: "Finally, it prints each word and its corresponding count." and an example output:

```
Word Frequency (
a: 2
is: 2
sample: 2
text: 2
this: 2
just: 1
```

## Few shot:

check whether the given word is palindrome or not ignoring punctuation and case by asking user dynamically.Example1.:a'mma output:It's a palindrome 2:akma: output:it's not a palindrome.



## Observation:

\*Zero-shot prompting gives the model only a task description. While it understands the basic goal (checking for palindromes), it often misses nuances like ignoring punctuation or formatting the output in a user-friendly way. The model may produce a technically correct function, but not one that matches the desired behavior or output style.

- Few-shot prompting, on the other hand, provides concrete examples. These examples act like a mini training session, showing the model:
  - How to clean the input (e.g., removing ' from "a'mma").
  - How to handle case sensitivity (e.g., "Akma" vs "akma").
  - What the expected output phrasing should be ("It's a palindrome" vs "It's not a palindrome").
- As a result, few-shot prompting leads to more accurate, consistent, and human-like responses. The model learns not just *what* to do, but *how* to do it—especially when dealing with edge cases or noisy input.

## Task Description#5

- Use few-shot prompting with 3 sample inputs to generate a function that determines the maximum of three numbers without using the built-in `max()` function.

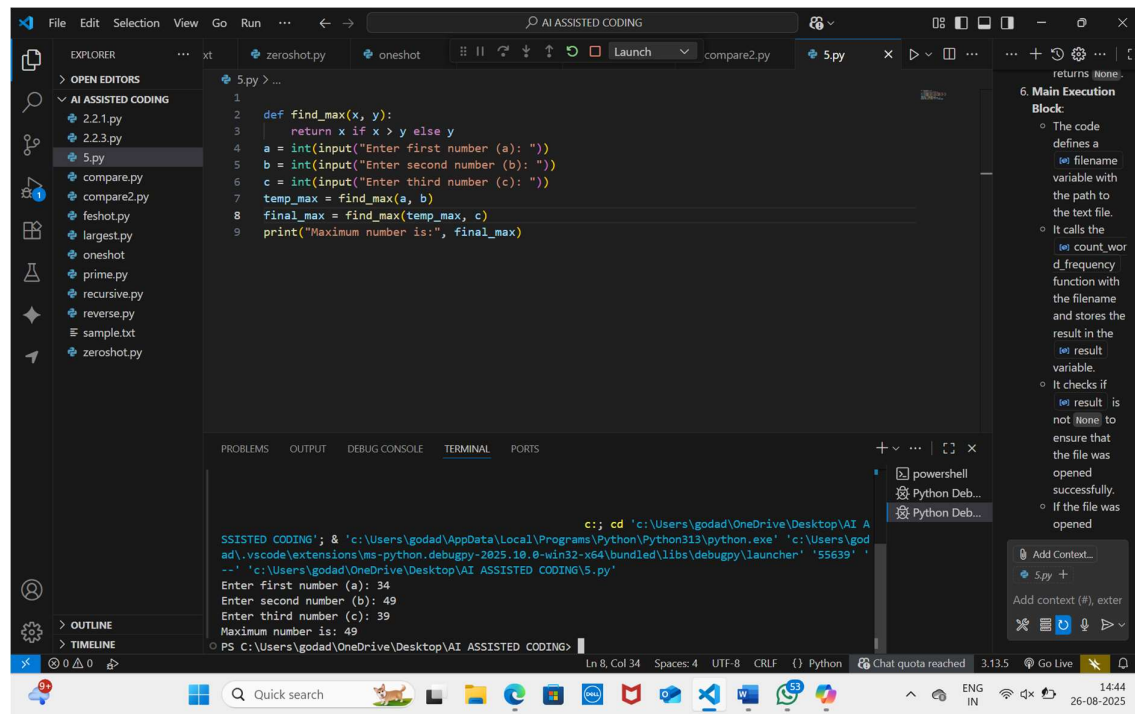
## Expected Output#5

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

### Prompt:

Determine the maximum of three numbers use other function in place of built-in max() function. Ask the user to input values dynamically. Example 1: numbers: a=10, b=20, c=50 → output: 50 Example 2: numbers: a=100, b=200, c=500 → output: 500 Example 3: numbers: a=1000, b=2000, c=5000 → output: 5000

### Code&Output:



The screenshot shows a Visual Studio Code editor window with a Python file named `5.py` open. The code defines a `find_max` function and uses it to find the maximum of three user-input numbers. The terminal at the bottom shows the execution of the script, where the user has entered 34, 49, and 39, resulting in the output 'Maximum number is: 49'.

```
1 def find_max(x, y):
2     return x if x > y else y
3
4 a = int(input("Enter first number (a): "))
5 b = int(input("Enter second number (b): "))
6 c = int(input("Enter third number (c): "))
7 temp_max = find_max(a, b)
8 final_max = find_max(temp_max, c)
9 print("Maximum number is:", final_max)
```

Terminal Output:

```
c;; cd 'c:\Users\godad\OneDrive\Desktop\AI ASSISTED CODING'; & 'c:\Users\godad\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\godad\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '55639' '--' 'c:\Users\godad\OneDrive\Desktop\AI ASSISTED CODING\5.py'
Enter first number (a): 34
Enter second number (b): 49
Enter third number (c): 39
Maximum number is: 49
```