

# Hepsiba Devara

2403A51L25

B-51

## **ASSIGNMENT -2.2**

## Task 1: Cleaning Sensor Data

**PROMPT:** Create a Python function that removes all negative values from a list of sensor readings.

```
File Edit Selection View Go Run ... ← → Q. AI ASSISTED CODING
EXPLORER ...
AI ASSISTED CODING
ASSIGNMENTS
ASSIGN-1-3.py
ASSIGN-2-2.py
DAY-12.py
LAB-2.py
LAB(2)PRACTICE SESSION...
Prime.py

ASSIGN-2-2.py ●
=====
1 """Create a Python function that removes all negative values from a list of sensor readings."""
2 def filter_negative_numbers(sensor_data):
3     ...
4     filtered_data = [x for x in sensor_data if x >= 0]
5     ...
6     return filtered_data
7
8
9 sensor_readings = [12, -5, 8, -3, 15, 0, -1]
10
11 print("Before filtering:", sensor_readings)
12 cleaned_readings = filter_negative_numbers(sensor_readings)
13 print("After filtering:", cleaned_readings)

BLACKBOX ... + ⚙️ ...
```

Build with Agent  
All responses may be inaccurate.  
Generate Agent

## **OUTPUT:**

The screenshot shows the VS Code interface with the Python extension installed. The top bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and AUGMENT NEXT EDIT. The TERMINAL tab is active, displaying command-line output for a script named 'ASSIGN-2-2.py'. The script uses the Python core library's random module to generate a list of integers and then filters it based on specific criteria. The bottom left pane shows the BREAKPOINTS view, which lists three types of exceptions: Raised Excepti..., Uncought Exce..., and User Uncought.... The bottom right pane shows a preview of the 'ASSIGN-2-2.py' file and a 'Describe what to build...' input field. The status bar at the bottom provides information about the current file (Ln 10, Col 1), encoding (UTF-8), and other settings.

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53462' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
Before filtering: [12, -5, 8, -3, 15, 0, -1]
After filtering: [12, 8, 15, 0]
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

**BREAKPOINTS**

- Raised Excepti...
- Uncought Exce...
- User Uncought...

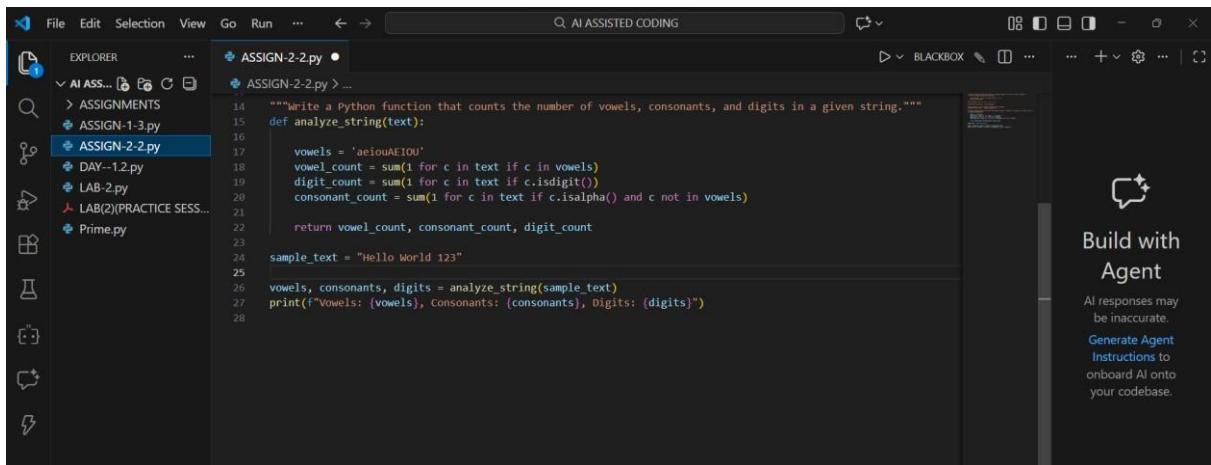
LN 10, COL 1 SPACES: 4 UTF-8 CRLF { } PYTHON ⏎ 3.14.2 ⏵ GO LIVE ⏵ BLACKBOXAI: OPEN CHAT ⏵ AUGMENT

## **EXPLANATION:**

This function removes invalid negative sensor values using list comprehension. Only values greater than or equal to zero are retained, ensuring clean IoT sensor data.

## Task 2: String Character Analysis

**PROMPT:** Write a Python function that counts the number of vowels, consonants, and digits in a given string.

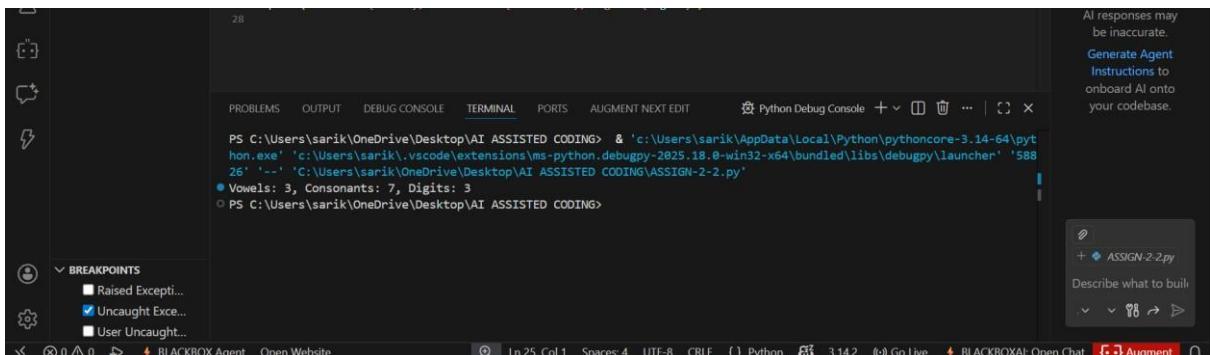


The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a folder structure with files like `ASSIGN-1-3.py`, `ASSIGN-2-2.py`, `DAY--12.py`, `LAB(2).PY`, and `Prime.py`.
- Code Editor:** Displays the Python code for the assignment:

```
14     """Write a Python function that counts the number of vowels, consonants, and digits in a given string."""
15     def analyze_string(text):
16         vowels = 'aeiouAEIOU'
17         vowel_count = sum(1 for c in text if c in vowels)
18         digit_count = sum(1 for c in text if c.isdigit())
19         consonant_count = sum(1 for c in text if c.isalpha() and c not in vowels)
20
21         return vowel_count, consonant_count, digit_count
22
23
24     sample_text = "Hello World 123"
25
26     vowels, consonants, digits = analyze_string(sample_text)
27     print(f"Vowels: {vowels}, Consonants: {consonants}, Digits: {digits}")
```
- AI Assistant Panel:** On the right, there's a panel titled "Build with Agent" with the sub-instruction "Generate Agent Instructions to onboard AI onto your codebase".

**OUTPUT:**



The terminal output in VS Code shows the following:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.8.0-win32-x64\bundled\libs\debugpy\launcher' '58826' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
● Vowels: 3, Consonants: 7, Digits: 3
○ PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

**EXPLANATION:**

The function iterates through each character and classifies it as a vowel, consonant, or digit.

Python string methods like `isalpha()` and `isdigit()` improve accuracy and readability.

### Task 3: Palindrome Check – Tool Comparison

**Gemini Prompt:** Write a Python function to check if a string is a palindrome. Ignore spaces and capitalization.

The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** AI ASSISTED CODING
- Left Sidebar:** RUN AND DEBUG, RUN, Run and Debug, To customize Run and Debug create a launch.json file, Debug using a terminal command or in an interactive chat.
- Code Editor:** File: ASSIGN-2-2.py  
21: return vowel\_count, consonant\_count, digit\_count  
22:  
23:  
24: sample\_text = "Hello World 123"  
25:  
26: vowels, consonants, digits = analyze\_string(sample\_text)  
27: print(f"Vowels: {vowels}, Consonants: {consonants}, Digits: {digits}")  
28:  
29: ##Gemini Prompt:  
30: #Write a Python function to check if a string is a palindrome. Ignore spaces and capitalization.  
31: def is\_palindrome\_gemini(s):  
32: s = s.replace(" ", "").lower()  
33: return s == s[::-1]  
34:  
35:  
36: print(is\_palindrome\_gemini("Racecar")) | True  
37:  
38: #Copilot Prompt:  
39: #Write a Python function to check palindrome. Consider only letters and ignore case.  
40: def is\_palindrome\_copilot(text):  
41: cleaned\_text = ''.join(c.lower() for c in text if c.isalnum())  
42: return cleaned\_text == cleaned\_text[::-1]
- Right Sidebar:** Build with Agent (AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.)

## OUTPUT:

The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** AI ASSISTED CODING
- Left Sidebar:** CALL STACK (Running), Breakpoints (Raised Except..., Uncaught Except..., User Uncaught...)
- Code Editor:** File: ASSIGN-2-2.py  
34: return s == s[::-1]  
35:  
36: print(is\_palindrome\_gemini("Racecar"))  
37:  
38: #Copilot Prompt:  
39: #Write a Python function to check palindrome. Consider only letters and ignore case.  
40: def is\_palindrome\_copilot(text):  
41: cleaned\_text = ''.join(c.lower() for c in text if c.isalnum())  
42: return cleaned\_text == cleaned\_text[::-1]  
43:  
44:  
45: print(is\_palindrome\_copilot("Racecar"))
- Terminal:** PS C:\Users\srrik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\srrik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\srrik\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58783' --- 'C:\Users\srrik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'  
True
- Right Sidebar:** AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.

**Copilot Prompt:** Write a Python function to check palindrome. Consider only letters and ignore case.

The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** AI ASSISTED CODING
- Left Sidebar:** RUN AND DEBUG, RUN, Run and Debug, To customize Run and Debug create a launch.json file, Debug using a terminal command or in an interactive chat.
- Code Editor:** File: ASSIGN-2-2.py  
37: #Copilot Prompt:  
38: #Write a Python function to check palindrome. Consider only letters and ignore case.  
39:  
40: def is\_palindrome\_copilot(text):  
41: cleaned\_text = ''.join(c.lower() for c in text if c.isalnum())  
42: return cleaned\_text == cleaned\_text[::-1]  
43:  
44:  
45: print(is\_palindrome\_copilot("Racecar"))
- Right Sidebar:** Build with Agent (AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.)

## OUTPUT:

The screenshot shows the Visual Studio Code interface with the following elements:

- Terminal:** Displays command-line output for Python 3.14.2. It shows the command `python.exe "c:/Users/sarik/OneDrive/Desktop/AI ASSISTED CODING/ASSIGN-2-2.py"`, followed by a True result and a note about AI assisted coding.
- Code Editor:** Shows a file named "ASSIGN-2-2.py".
- AI Agent Sidebar:** Located on the right, it says "Agent" and "AI responses may be inaccurate." It includes a "Generate Agent Instructions" button and a note about onboard AI onto your codebase.
- Breakpoints:** A sidebar panel shows breakpoints: "Raised Excepti..." (unchecked), "Uncaught Exce..." (checked), and "User Uncaught..." (unchecked).
- Bottom Status Bar:** Shows file path "C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py", line 45, column 42, spaces: 4, encoding: UTF-8, Python 3.14.2, Go Live, BLACKBOXAI: Open Chat, and Augment.

## Comparison Table:

Feature	Gemini	Copilot
Clarity	Simple, minimal code	Slightly longer, more robust
Handling spaces/case	Ignores spaces, converts to lowercase	Ignores spaces and punctuation, lowercase
Readability	Very clear	Clear, slightly more detailed
Efficiency	Uses string slicing	Uses string comprehension

### EXPLANATION:

Gemini provides concise and easy-to-read logic, making it beginnerfriendly. Copilot generates more robust code that handles punctuation and special characters.

## Task 4: Code Explanation Using AI Step 1 –

### Code Snippet:

The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar with icons for file operations like Open, Save, and Run, along with sections for RUN AND DEBUG and RUN. The main area displays a Python file named 'ASSIGN-2-2.py'. The code contains two lines of Python code:

```
49     ##Step 1 - Code Snippet(Code Explanation)
50     def is_palindrome(text):
51         text = text.replace(" ", "").lower() # Remove spaces and lowercase
52         return text == text[::-1]           # Compare string with its reverse
```

To the right of the code, there's a sidebar titled 'Build with Agent' which includes a note that AI responses may be inaccurate and a link to 'Generate Agent Instructions'.

## Step 2 – AI Explanation:

1. `text.replace(" ", "").lower()` → Removes spaces and converts letters to lowercase.
2. `text == text[::-1]` → Checks if the string is equal to its reverse.

### EXPLANATION:

The function normalizes the string to avoid case and space mismatches. It then compares the string with its reverse to verify palindrome logic.