PYTHON PROGRAMMING INTERNSHIP

(PROJECT BASED)

WEEK 2 PROJECT(PROJECT 2)

PROJECT: Word Counter.

REQUIREMENTS AND FEATURES OF PROJECT:

User Input: Prompt the user to enter a sentence or paragraph.

Word Counting Logic: Implement a function that counts the number of words in the

Output Display: Print the word count to the console.

Error Handling: Account for potential errors, such as empty input.

Code Comments: Add comments to explain the purpose of different parts of your code.

User-Friendly Interface: Ensure a clear and user-friendly interface for input and output.

Project step by step plan:

- **Input Handling**: Prompt the user to enter a sentence or paragraph.
- **String Manipulation**: Use string methods to process the text and identify word boundaries.
- **Function Creation**: Define a function to handle the word-counting logic.
- **Control Flow and Error Handling**: Check for empty input and handle it gracefully.
- **Output Display**: Print the final word count.

 Code Comments: Comment each part to clarify what it does.

 User-Friendly Interface: Make the interaction smooth and intuitive

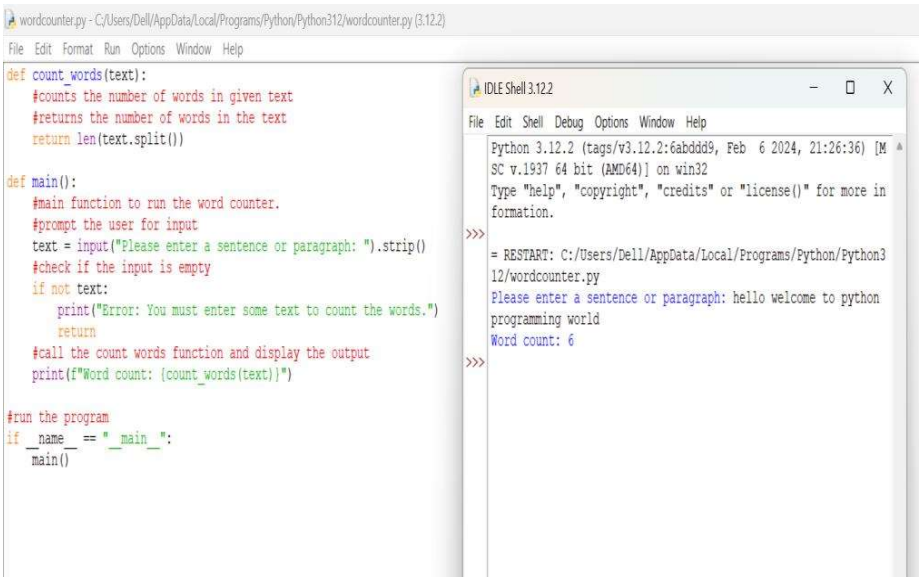
Code for the word conter:

```
def count_words(text):
    #counts the number of words in given text
    #returns the number of words in the text
    return len(text.split())


def main():
    #main function to run the word counter.
    #prompt the user for input
    text = input("Please enter a sentence or paragraph: ").strip()
    #check if the input is empty
    if not text:
        print("Error: You must enter some text to count the words.")
        return
    #call the count words function and display the output
    print(f"Word count: {count_words(text)}")
```

```
#run the program
if __name__ == "__main__":
    main()
```

code execution and output display:



Note:

Code executed in IDLE(python 3.12 64 bit) compiler.

Code Explanation

1. **count_words(text)**:

- This function accepts text as input and splits it into words using the .split() method, which splits the string by spaces.
- It counts the length of the resulting list to get the number of words.

2. **main()**:
   - Prompts the user for input and removes any leading or trailing spaces.
   - If the input is empty, it displays an error message.
   - Calls count_words to get the word count and prints it.
3. **Error Handling**:
   - Checks for empty input and handles it with an error message.
4. **User Interface**:
   - Displays a welcome message and clearly prompts the user for input.

Conclusion:

This project word counter gives good experience with input handling, string manipulation, and function creation in Python.

The above code takes the input text and gives output as number words are there in given text.

Submitted by:

Nayikam Bhanuprasad

Bv raju institute of technology

Email:23211a04g5@bvrit.ac.in