2025 ASEE Northeast Section Conference, March 22, 2025, University of Bridgeport, Bridgepeort, CT, USA.

On Teaching with AI: The Case of Scientific and Engineering Programming in the Era of Artificial Intelligence

Basile Panoutsopoulos
Department of Physics and Engineering
Community College of Rhode Island
Warwick, RI
BPanoutsopoulos@ccri.edu

Abstract—**In this paper, we present a background on the use of Artificial Intelligence (AI) in education and report on its application in a course in "Scientific and Engineering Programming in C++" is provided. Interaction with AI is provided based on the feedback from students-instructor discussions.**

Keywords—Artificial Intelligence, AI, Education, Programming, C++

I. INTRODUCTION

Artificial Intelligence (AI) is a reality in the current era with accelerated use in the educational world. Its use transforms the academic field from the classroom to the study room by providing a more personalized environment.

Artificial intelligence (AI) refers to the capability of computational systems to perform tasks typically associated with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making. It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals [1].

Similarly, artificial intelligence (AI) is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings [2].

Education is one of the fields where AI has a great impact. Education tends to adopt various modern practices to improve the overall educational experience, student engagement, and lesson planning. Various services and tools based on artificial intelligence are already used in the educational process. Some benefits from the use of AI in education are:

Personalization: AI can give us a differentiated and personalized learning experience. For example, it can analyze students' strengths and interests, track progress and performance, and identify areas in which they are not successful.

Customization: AI-based tools can customize the content and delivery method and adapt to the speed of each student so that they can learn differently and at their own pace. For example, Duolingo is a fun, free app for learning 40+ languages through quick, bite-sized lessons. Practice speaking, reading, listening, and writing to build your vocabulary. [3]

Inclusion: AI can improve accessibility and ensure that students with disabilities can access rich learning opportunities. For example, AI-powered tools can create subtitles and translations of a presentation that is taking place in real-time.

Active participation: AI can provide engaging learning experiences. For example, virtual and augmented reality technologies allow for the creation of interactive and immersive learning environments where students can explore and interact with content more engagingly and memorably.

Automation: AI can automate administrative tasks, such as grading, to free up teachers' time to focus on other important aspects of teaching.

In this work, the case of a course on "Scientific and Engineering Programming Using C++" is examined. It is a transformation from a typical programming course to a course where the use of AI is welcomed and encouraged.

AI can provide pseudocode and flowcharts. The text mode flowchart was fine, but the graphical one was a disaster. AI can provide computer programs that are logically and syntactically correct. It can add comments to each line of code. The provided explanatory tutorial statements are great for understanding the material. We may notice that a comment error was discovered on a simple program of adding two numbers and slightly different outputs based on the wording of the input question. Although the example used is simple (adding two numbers), real problems are complex. Solving correctly and managing the development of code is a complicated process. Teaching skills such as Top-Down Design becomes necessary even in an introductory first course. "This is a skill [problem decomposition] to know early on because you need to break a large problem into smaller pieces that an LLM [Large Language Model] can solve," says Leo Porter, an associate teaching professor of computer science at the University of California, San Diego. Now, it becomes a priority in introductory classes" [4].

Technology is to be used for the benefit of humans. But as is always the case, technology can be used positively or negatively, ethically or unethically. The goal of this programming course, as with every other course, is to educate the students to fulfill the course's outcomes successfully. They say, "If you can't beat them, join them." This seems to be the ideal case to apply this saying here.

II. USE OF AI IN THE COURSE

A. *The Proposal*

We shall avoid proposing to exclude AI from the course or much more from the curriculum. This would be something out of anybody's control, much more of the instructor. On the contrary, we propose an approach to incorporate AI in the course in a beneficial way. At the same time, we must ensure that the student studies the material and develops critical and creative thinking skills. "Students are early adopters and have been actively testing these tools," says Johnny Chang, a teaching assistant at Stanford University pursuing a master's degree in computer science [4]. By learning how to use the available AI tools and resources, teachers can get the confidence to navigate the challenges and seize the opportunities present and opening in this era [5].

B. *The Process*

The proposed approach lies around the central theme that the use of AI is welcomed and encouraged. The approach has been tested twice and as with every active course, it will continue to be tested and refined this semester for the third time. It looks like the students are using it responsibly and beneficially. They are required to comment on every line of the code (something that AI can do well), but here are requirements on the organization of the assignment as part of the code file: Title Page, Statement of the Problem, Definition of each variable, organization of the program using functions, comment every function, use keywords that they have been covered in the course, describe the output, present, input and output in a user-friendly format (the course does not use windows interface but rather a command line approach (It reminds DOS) etc. In the end, the solution to the problems has been typically organized as a "tutorial" process; it provides a guided learning experience for the student, with clear, sequential steps to complete a task and understand the concepts.

A weekly short examination reinforces the material and confirms the student's competence. The students are embracing the new approach of AI to help in their first steps of programming and understanding complex concepts, as many aspects of programming include learning how to code.

III. A SESSION WITH AI

In this work, we examine the case of a course on "Scientific and Engineering Programming Using C++." It has been transformed from a typical programming course to a course where the use of AI is welcomed and encouraged. AI can provide pseudocode, flowcharting, and computer programs that are logically and syntactically correct; it can add comments to each line of code and provide explanatory tutorial statements. The user asked a simple question as shown in Fig 1 and the corresponding answer is shown in Fig. 2.

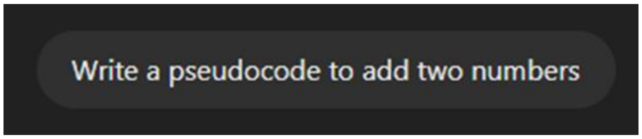
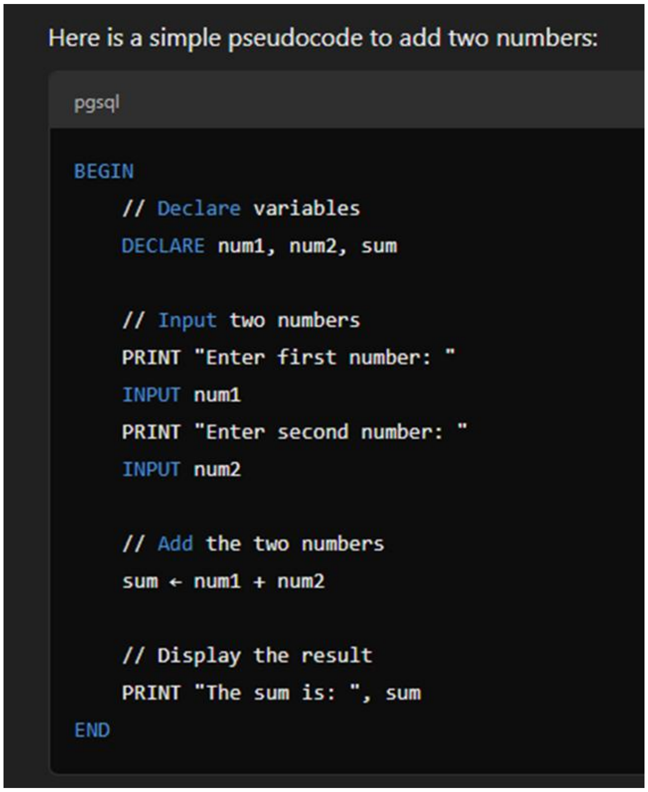

Write a pseudocode to add two numbers

Fig. 1. Interaction of use AI.


Here is a simple pseudocode to add two numbers:

```
BEGIN
    // Declare variables
    DECLARE num1, num2, sum

    // Input two numbers
    PRINT "Enter first number: "
    INPUT num1
    PRINT "Enter second number: "
    INPUT num2

    // Add the two numbers
    sum ← num1 + num2

    // Display the result
    PRINT "The sum is: ", sum
END
```

Fig. 2. Provided pseudocode.

Whereas an explanatory note is noted in Fig. 3.

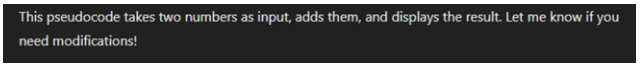
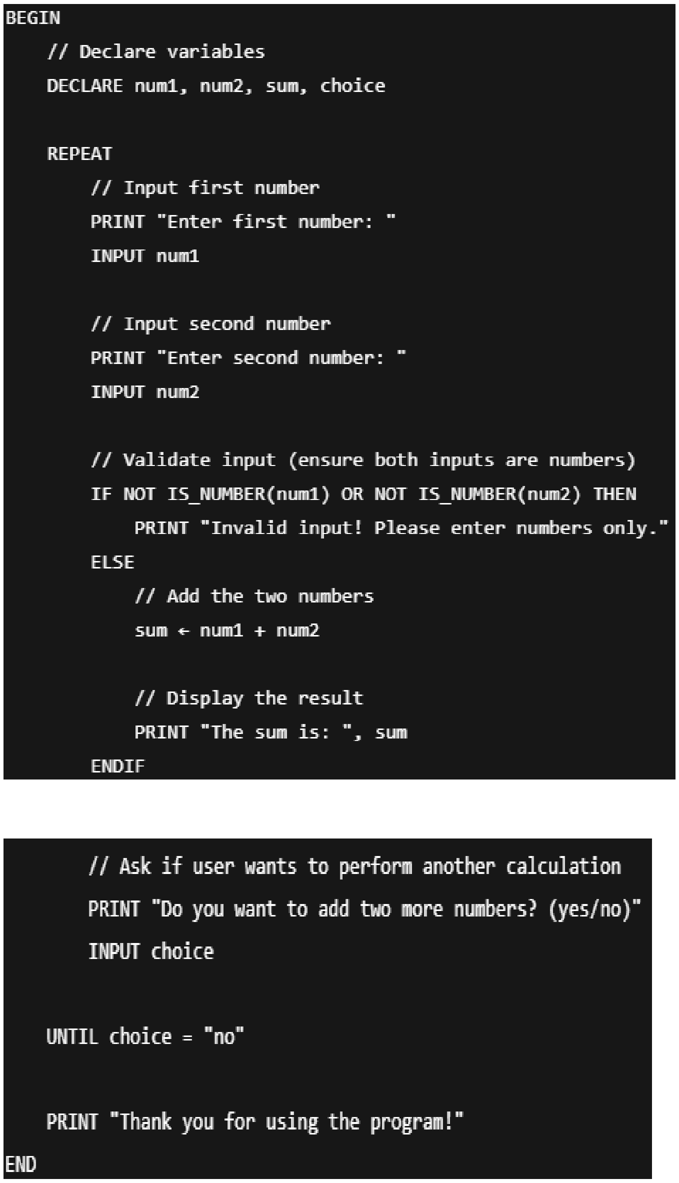

This pseudocode takes two numbers as input, adds them, and displays the result. Let me know if you need modifications!

Fig. 3. Explanation of the pseudocode along with the invitation for new input of Fig 2.

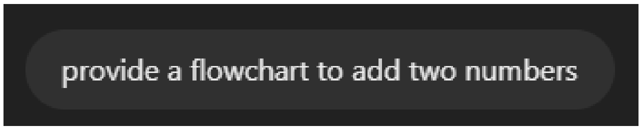
The comment " Your pseudocode looks great! Do you need any modifications, like handling invalid inputs or allowing multiple calculations in a loop?" the "Yes" reply resulted in a modified code able to handle invalid input shown in Fig. 4.



```
BEGIN
    // Declare variables
    DECLARE num1, num2, sum, choice

    REPEAT
        // Input first number
        PRINT "Enter first number: "
        INPUT num1

        // Input second number
        PRINT "Enter second number: "
        INPUT num2

        // Validate input (ensure both inputs are numbers)
        IF NOT IS_NUMBER(num1) OR NOT IS_NUMBER(num2) THEN
            PRINT "Invalid input! Please enter numbers only."
        ELSE
            // Add the two numbers
            sum ← num1 + num2

            // Display the result
            PRINT "The sum is: ", sum
        ENDIF

        // Ask if user wants to perform another calculation
        PRINT "Do you want to add two more numbers? (yes/no)"
        INPUT choice

    UNTIL choice = "no"

    PRINT "Thank you for using the program!"
END
```

Fig. 4. Improved pseudocode to handle invalid input.

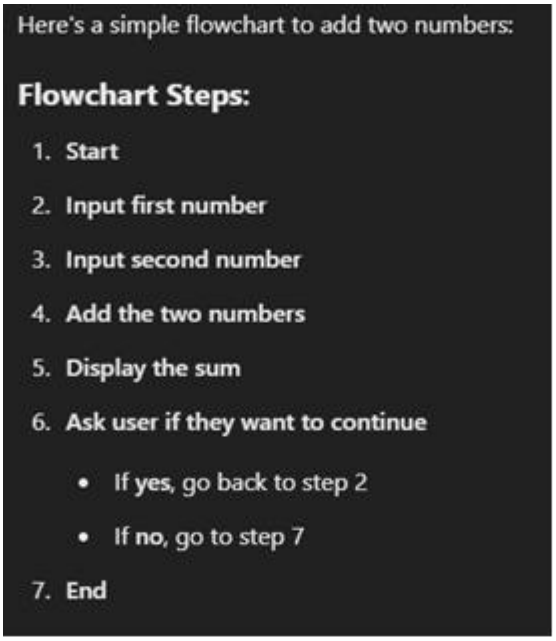
Then, the interactive session of human-AI continued with a request for a flowchart is shown in Fig. 5.



provide a flowchart to add two numbers

Fig. 5. Request for a flow chart.

The AI reply came as flowchart steps are shown in Fig 6.



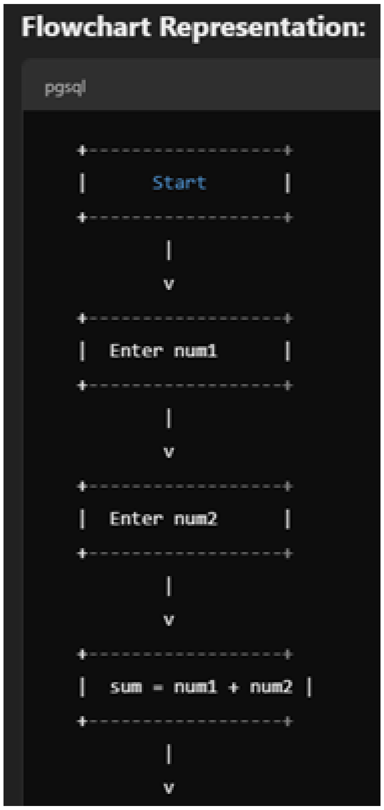
Here's a simple flowchart to add two numbers:

Flowchart Steps:

1. Start
2. Input first number
3. Input second number
4. Add the two numbers
5. Display the sum
6. Ask user if they want to continue
    - If yes, go back to step 2
    - If no, go to step 7
7. End

Fig 6. The provided flowchart steps.

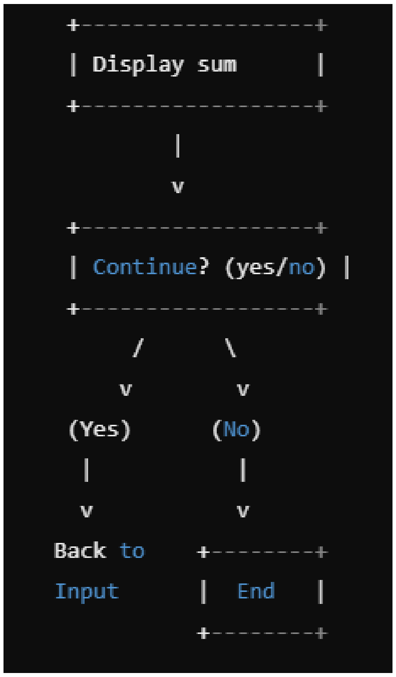
At the same time, the flowchart representation was provided in Fig. 7a and Fig. 7b.



Flowchart Representation:

```
+------------------+
|      Start       |
+------------------+
         |
         v
+------------------+
|  Enter num1      |
+------------------+
         |
         v
+------------------+
|  Enter num2      |
+------------------+
         |
         v
+------------------+
|  sum = num1 + num2  |
+------------------+
         |
         v
```

Fig 7a. The first part of the flowchart representation.



```
+------------------+
| Display sum      |
+------------------+
         |
         v
+------------------+
| Continue? (yes/no) |
+------------------+
      /       \
     v         v
 (Yes)       (No)
   |           |
   v           v
 Back to    +--------+
 Input      |  End   |
            +--------+
```

Fig 7b. The second part of the flowchart representation.

The previous session ended with another question from AI: “Would you like a visual version of the flowchart? I can generate an image for you!”, Fig. 8.

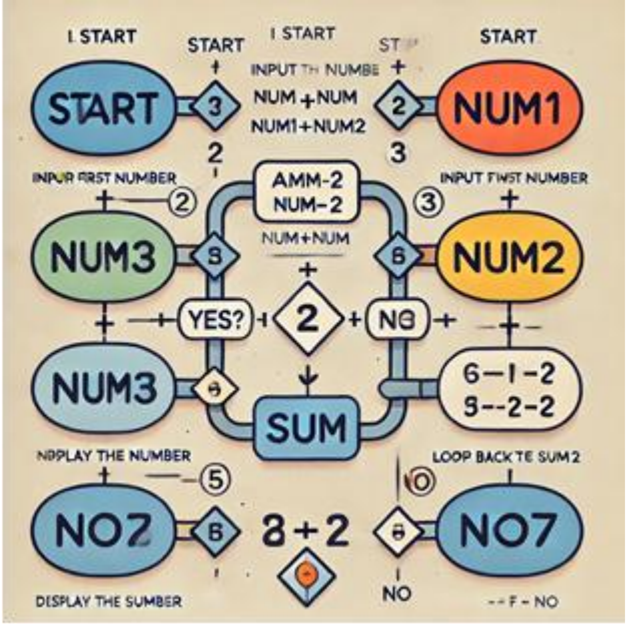


Fig. 8. The provided visual version of the flowchart with the comment: “ Here is the flowchart for adding two numbers. Let me know if you need any modifications or a more detailed version!”

There is something wrong here. This is not a user-friendly representation!

The session continued with the invitation: “What modifications would you like? Do you want more details, a different style, or additional steps like input validation? Let me know how you'd like it improved!”

With the user request: “ better clearer flowchart”. The reply is provided on Fig. 9.

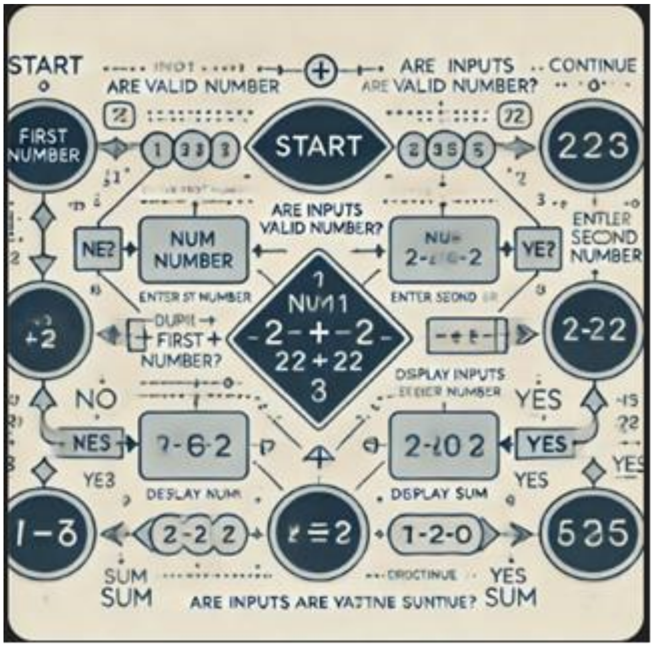
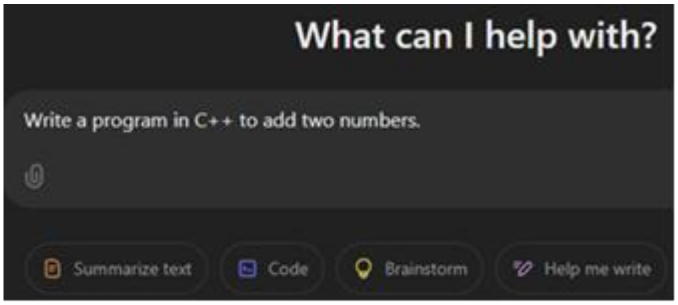


Fig. 9. The improved version of the visual flowchart.

The improved version of the visual flowchart, Fig. 9, was accompanied by the comment: “Here is a clearer and more structured flowchart for adding two numbers. Let me know if you need any further refinements!”

The next part of the interaction is a request for the programming code. The code must be implemented in C++, Fig. 10.

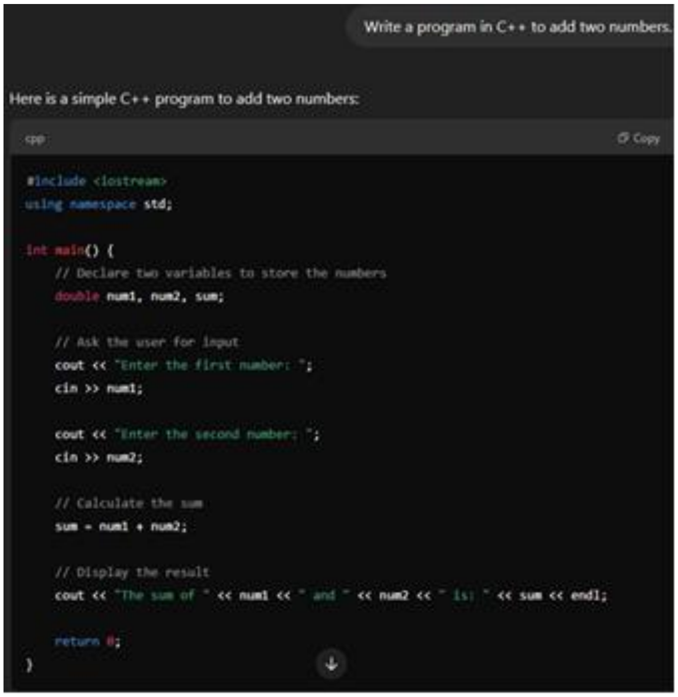


What can I help with?

Write a program in C++ to add two numbers.

Fig. 10. The request for programming code.

The request in this interactive session came with a reply shown in Fig. 11. The comment “//Declare two variables to store the numbers” while in the next line, three numbers are declared is a topic for discussion.



Write a program in C++ to add two numbers.

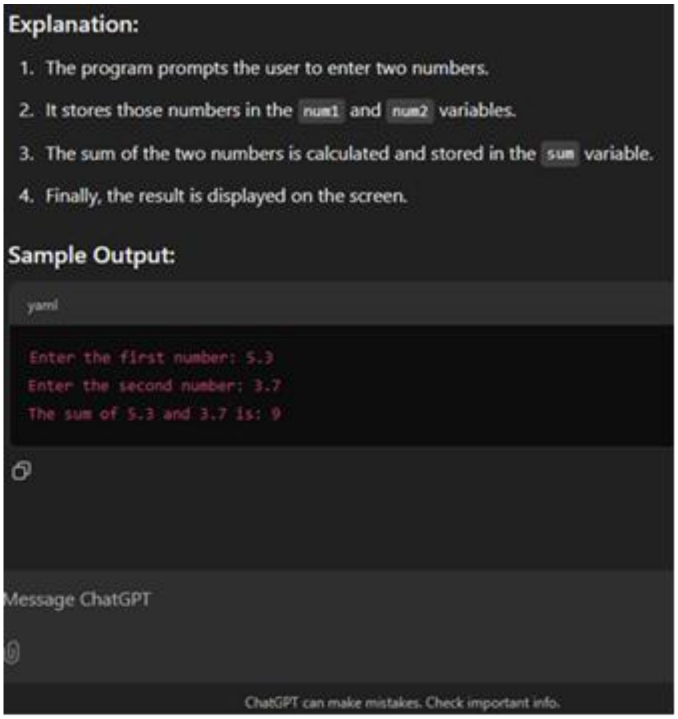
Here is a simple C++ program to add two numbers:

```cpp
#include <iostream>
using namespace std;

int main() {
    // Declare two variables to store the numbers
    double num1, num2, sum;

    // Ask the user for input
    cout << "Enter the first number: ";
    cin >> num1;

    cout << "Enter the second number: ";
    cin >> num2;

    // Calculate the sum
    sum = num1 + num2;

    // Display the result
    cout << "The sum of " << num1 << " and " << num2 << " is: " << sum << endl;

    return 0;
}
```

Fig. 11. The programming code in C++.

The “Explanations” of the code are followed in Fig. 12 at the top. We believe that this specific part is a valuable tool for understanding the programming process. It can be used as a recitation and for self-study.

The next session is the “Sample Output” in Fig. 12 at the bottom. This is the final interactive result between the program and a human or machine which was altogether the goal of this exercise.



Explanation:

1. The program prompts the user to enter two numbers.
2. It stores those numbers in the `num1` and `num2` variables.
3. The sum of the two numbers is calculated and stored in the `sum` variable.
4. Finally, the result is displayed on the screen.

Sample Output:

```
Enter the first number: 5.3
Enter the second number: 3.7
The sum of 5.3 and 3.7 is: 9
```

Fig. 12. The “Explanations” of the code and “Sample Outputs”.

It is suggested that the reader interface and interact with one or more AI engines, repeating the above process. Observing and critically discussing the results will be a valuable part of the process.

FURTHER WORK

A complete AI-driven course will be of interest to design and implement. Other topics for further study may include ethical concerns and legal issues related to the ownership of the material presented. There are cases in which AI cannot be used, for example, in cases of classified work.

REFERENCES

[1] [1] Russell, Stuart J.; Norvig, Peter. Artificial Intelligence: A Modern Approach (4th ed.). Hoboken: Pearson. 2021.

[2] https://www.britannica.com/technology/artificial-intelligence [Last accessed 3rd March 2025]

[3] https://www.duolingo.com/ [Last accessed 3rd March 2025]

[4] AI Copilots Are Changing How Coding Is Taught. https://spectrum.ieee.org/ai-coding [Last accessed 3rd March 2025]

[5] Jose Antonio Bowen and C. Edward Watson. Teaching with AI. A Practical Guide to A New Era of Human Learning. Johns Hopkins University Press. 2024.