The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #42963

Teaching Programming Languages by Two Teachers: Instructor and ChatGPT

Dr. Alireza Kavianpour, DeVry University

Dr. Alireza Kavianpour received his PH.D. Degree from University of Southern California (USC). He is currently Senior Professor at DeVry University, Ontario, CA. Dr. Kavianpour is the author and co-author of over sixty technical papers most of them published in IEEE journals or conferances. He has one pending patent .

©American Society for Engineering Education, 2024

Teaching Programming Languages by Two Teachers: Instructor and ChatGPT

This paper discusses how an instructor can enhance teaching an undergraduate programming course, specifically C++ programming language, utilizing Artificial Intelligence (AI) tools, such as ChatGPT.

The teaching experience and a collection of data from a C++ programming course, with and without ChatGPT, will be analyzed. Several guidelines to use AI tools more efficiently for a programming course will be presented. In this programming course, homework assignments were modified to address the use of ChatGPT. Most undergraduate engineering students must take this programming course with Calculus course as a prerequisite.

A neurological study has shown that "curiosity makes people more receptive for learning, and that as we learn, we enjoy the sensation of learning". That is exactly what the author noticed among students in discovering the use of this new tool.

Overall, based on the author's experiences, most students were **curious** to learn more about this new tool. As a result, they were more engaged in writing codes. The author has warned students that scientific advances can both help and hurt and AI is no exception. Students highly welcomed this new tool and were eager to use it in a positive way in the other courses to enhance their learning.

(1) Introduction

There is a lot of buzz around AI and specifically, how ChatGPT[6,7] is being used in the academic field. Most notable comments include: "AI will take over the role of an instructor" or "students will never write their own research papers or their own programming code."

Several years ago, when calculators were first introduced, many mathematicians and teachers opposed the use of them, and claimed students would never learn mathematics with such a tool available to them. As time passed, calculators have been proven to enhance a student's mathematical learning capabilities. As such, like any new technology, teachers and students need to learn the right way to use it.

A programming textbook has a limited capability to explain a topic as explained in references1,2, and 3. Students can find out more about a topic from an AI generator than searching through countless pages on the internet. Students can get clear and readable responses that can help them to view the material from a different perspective.

AI tools are designed by using machine learning algorithms known as the transformer model. This model is explained in references 4 and 5. This model uses a vast majority of big data, books, articles and generates a response. The more data results, the better the response will be. The one aspect that makes AI stand out from other search engines, such as Google, is that you can interact and ask follow-up questions. This is a very useful feature to give students more practice by asking them to challenge AI on specific topics.

In any computer programming course, an instructor must teach students the foundation of programming and swiftly bring them to a level of skill. Students can use computing methods in their courses to do some of the following: analyzing data, learning linear algebra, solving problems in mechanics, solving differential equations, and use it in embedded programming. To align the teaching of programming courses with the use of ChatGPT, we must review and update teaching and course objectives. Teaching objectives in programming courses are as follows:
a-The main objective in teaching a computer language is to enable the students to convert engineering problems into an algorithm.
b-The conversion of this algorithm into a program written in one programming language is of secondary importance.
c-Consequently, the choice of the teaching language should be governed by which language provides the best support to the student in performing the implementation part of the problem.
Outcomes in programming courses are as follows:
a-To be able to design, develop, compile, and debug programs in a high-level programming language.
b-To be able to develop programs for solving numerical problems applicable to engineering.
c-Express data of multiple types (numbers, strings, lists) in a high-level language.
d-Write expressions and functions in a high-level language statement.
e-Use control flow constructs to combine multiple statements into a larger subprogram.
f-Test and debug programs with I/O constructs.

(2) AI Implementation in a Programming Course

After teaching computer programming for many years, this past year, I decided to teach one of the freshman courses by using AI to explain different concepts in computer programming; in particular C/C++ programming. AI was used for topics such as: control statements (*if, while, do while, switch,* and *for*), functions, pointers, structures, file handling, and generating a document for a program. I found that these concepts and other topics could be taught more efficiently using AI and it would reduce teaching time. For the students' homework assignments, I implemented a 2-step assignment. First, I asked students to complete a set of questions on their own. Second, I asked them to use ChatGPT to check if they missed anything. An example question:
"Write a program on sorting fifty numbers. Then, ask ChatGPT to offer edits and suggestions. Now, compare the two, and tell why ChatGPT made those changes?"
Several homework assignments in programming with the use of ChatGPT are presented below.

(a) Teaching the Concept of Integer and Float Point Numbers
Homework 1:
For the following assignment, calculate the result.
(i) A=3*5/3-3
(ii) B=3*5.0/3-3
(iii) Ask ChatGPT why results are different.

```
#include <stdio.h>
int main()
{int A=4*5/3-3;
float B= 4*5.0/3-3;
printf("%d\n%f", A,B);
return 0;}
```

Answer:

```
3
3.666667
```

Question to ChatGPT:
Explain why answers for these two calculations are different.
(i) A=4*5/3-3
(ii) B=4*5.0/3-3

Answer from ChatGPT (without editing):
"The difference in the results of the two calculations is due to the way numerical operations are handled with different data types in programming or mathematical

expressions. In B expression, the only difference is that 5.0 is a floating-point number, which means it has decimal precision. When you perform the multiplication (4 * 5.0), the result is 20.0. Then, when you divide 20.0 by 3, you get approximately 6.666667. Finally, subtracting 3 from 6.666667 gives you the result: B ≈ 6.666667 - 3 ≈ 3.666667. So, the difference lies in the data type and precision of the numbers involved in the calculations. In the first case, where all numbers are integers, the result is an integer (3), while in the second case, where one of the numbers is a floating-point number, the result is a floating-point number (3.666667)”.

ChatGPT explains it nicely and saves the teaching time so the teacher could spend more time on the other topics.

(b) Teaching the Concept of Assignment and Comparing Commands
Homework 2:
For the following program, calculate the result.
(i) if(A=1)
(ii) if(A==1)
(iii) Ask ChatGPT why answers are different.

```
#include <stdio.h>
int main()
{int A=0;
if(A=1)
printf("Answer=%d",A);
else
printf("Answer=%d",A);
return 0;}
```

```
Answer=1
```

```
#include <stdio.h>
int main()
{int A=0;
if(A==1)
printf("Answer=%d",A);
else
printf("Answer=%d",A);
return 0;}
```

```
Answer=0
```

Question to ChatGPT:
What is the difference between ***if(A=1)*** *and* ***if(A==1****) in C programming?*

Answer from ChatGPT (without editing):
"In C programming, the difference between ***If(A=1)*** and ***If(A==1)*** is crucial and related to the use of assignment and equality operators.

If(A=1):

This expression uses a single equal sign, which is the assignment operator. It assigns the value 1 to the variable or expression A. The condition in the ***if*** statement is the result of the assignment operation, which is the assigned value (1 in this case).

The ***if*** statement will evaluate the truthiness of the assigned value, and the block of code inside the if statement will be executed if the assigned value is considered true.

If(A==1):

This expression uses a double equal sign, which is the equality operator.
It checks if the value of A is equal to 1. The condition in the if statement is the result of the equality comparison, which will be true if A is equal to 1, and false otherwise. The block of code inside the ***if*** statement will be executed if the condition is true. That is block of code will be executed only if the value of A is equal to 1.

In summary, ***If(A=1)*** is an assignment operation and will always evaluate to true (unless the assigned value is 0), while ***If(A==1)*** is a comparison operation checking whether the value of A is equal to 1. It is generally the equality operator ***(==)*** that you want to use when comparing values in conditional statements."

(c) Teaching the Concept of LOOPs
Homework 3:
Consider ***for (; ; ;)*** statement:
(i) Is it a correct statement?
(ii)Ask ChatGPT what is the use of ***for (; ; ;)*** in C programming.

Question to ChatGPT:
What is the use of ***for (; ; ;)*** *in C programming?*

Answer from ChatGPT (without editing):
"In C programming, ***for (; ; ;)*** is an infinite loop. It is a shorthand notation for a loop with no initialization, no condition, and no increment or decrement

expressions. In practice, infinite loops are used in situations where continuous execution is desired until the program is manually terminated. Care should be taken when using infinite loops, as they can lead to programs without exit unless forcefully stopped, and this can result in an unresponsive or hung program. Infinite loops are often used in embedded systems, real-time applications, or situations where continuous execution is required."

(d) Teaching the Concept of Pseudo Instructions
Homework 4
Explain the difference between ***#define a 2*** and ***int a=2*** although both means ***a=2***.

Question to ChatGPT:
Explain the difference between **#define a 2** *and* **int a=2** *although both means* ***a=2.***

Answer from ChatGPT (without editing):
"The lines ***#define a 2*** and ***int a = 2***; serve different purposes in C programming and are used in different contexts: ***#define a 2*** is a preprocessor directive for text substitution, and it doesn't involve the allocation of memory. ***int a = 2***; is a variable declaration and definition that allocates memory to store an integer value and initializes it with the specified value."

(e) Teaching The Concept of Global and Local Variables
Homework 5:
Consider the following program:
(i) What is the difference?
(ii) Ask the same question from ChatGPT and compare your answer.

Question to ChatGPT:
In the following program, why variable ***z*** *is not 7?*

```
#include <stdio.h>
int z=3;
int f(void);
void main(void)
{int w=f(); printf("%d",z); }
```

int f(void)
{int z = 7; return 1; }

Answer from ChatGPT (without editing):
"In the provided program, the reason why ***z*** is not 7 when printing is due to the use of a local variable with the same name ***z*** inside the function ***f***.
In function ***f***, there is a local variable ***z*** declared with the value of 7. This local variable ***z*** shadows (hides) the global variable ***z*** declared outside the function. When the function ***f*** is called in ***main***, it returns 1, but the local variable ***z*** inside the function is not affecting the global variable ***z***. The global ***z*** remains 3, and that's what gets printed in the ***printf*** statement in ***main***. If you want the function ***f*** to modify the global variable ***z,*** you should remove the redeclaration of ***z*** inside the function."

(f) Teaching the Concept of Static Variables
Homework 6:
Consider ***int a*** and ***static int a*** statements:
(i) What is the difference?
(ii)Ask the same question from ChatGPT and compare your answer.

Question to ChatGPT:
Explain the difference between ***int a*** *and* ***static int a****.*

Answer from ChatGPT (without editing)
The difference between ***int a*** and ***static int a*** lies in the storage duration and scope of the variable in C or C++ programming.
int a:
Declares an integer variable named ***a***. The variable a has automatic storage duration by default.
static int a:
Declares a static integer variable named ***a***. The variable ***a*** has static storage duration. Static variables are created once the program starts and persist throughout the program's execution. The scope of ***a*** is limited to the block or function in which it is declared, but its lifetime extends to the entire program.
"In summary, the key difference is in the storage duration and lifetime of the variable. Automatic variables (declared without static) have a limited lifetime within their enclosing block, while static variables persist throughout the program's execution."

(h) Teaching the Concept of Pointers
Homework 7:
Consider ***int a*** and ***int * a*** statement:
(i) What is the difference?
(ii)Ask the same question from ChatGPT and compare your answer.

Question to ChatGPT:
Explain the difference between ***int a*** and ***int *a***

Answer from ChatGPT (without editing):
int a:
Declares a variable named a of type ***int.*** The variable ***a*** holds an integer value. You can directly store and manipulate integer values in the memory location allocated for **a**.
int *a:
Declares a variable named ***a*** of type "pointer" or simply, an integer pointer. The variable ***a*** does not directly store an integer value; instead, it stores the memory address (pointer) where an integer value is located. You need to allocate memory separately to store the integer value that ***a*** point to.
"In summary, ***int a*** declares a regular integer variable, while **int **a*** declares a pointer to an integer. The pointer is used to store memory addresses, allowing you to indirectly access and manipulate the value stored at that address."

(i) Teaching the Concept of Structures
Homework 8:
Consider *array* and *structure*:
(i) What is the difference?
(ii) Ask the same question to ChatGPT and compare your answer.

Question to ChatGPT:
Explain the difference between array and structure.

Answer from ChatGPT (without editing):

"In summary, the key differences between arrays and structures are:

(a) Arrays store elements of the same data type in contiguous memory.

(b) Structures group variables of different data types under a single name.
(c) Array elements are accessed using indices.
(d) Structure members are accessed using dot notation.

Additionally, you can create arrays of structures, combining the benefits of both concepts to represent collections of records with different data types.”

(j) Documentation of a Program

Homework 9:

(i) Write a document for the following program.
(ii) Ask ChatGPT to generate a document for the same program.
(iii) Compare your document with the one generated by ChatGPT. Did you miss any items?

Question to ChatGPT:

Write a document for the following program.

```
#include <stdio.h>
void main (){
  int num[100], i, j, a, n;
  printf("Input size of array");
  scanf("%d", &n);
  printf("Input numbers");
  for (i = 0; i < n; ++i)
    scanf("%d", &num[i]);
  for (i = 0; i < n; ++i){
    for (j = i + 1; j < n; ++j){
      if (num[i] > num[j]){
        a = num[i];
        num[i] = num[j];
        num[j] = a;}}}
  printf("Result is:");
  for (i = 0; i < n; ++i){
    printf("%d", num[i]); }}
```

Answer from ChatGPT (without editing):
“Sorting Program in C

Introduction
This document provides an explanation and overview of a simple C program designed to sort an array of integers in ascending order using the Bubble Sort algorithm.

Program Overview
The program begins by declaring an array **num** of size 100 and variables **i, j, a,** and **n** to handle array elements, loop counters, a temporary variable, and the number of elements, respectively.

User Input
The program prompts the user to enter the number of elements in the array and then takes the input values for each element.

Sorting using Bubble Sort
The sorting is performed using the Bubble Sort algorithm, where pairs of adjacent elements are compared and swapped if they are in the wrong order. This process is repeated until the entire array is sorted.

Displaying the Sorted Array
Finally, the program prints the sorted array in ascending order.

Summary
The program efficiently sorts an array of integers in ascending order using the Bubble Sort algorithm. Understanding this program provides insight into basic array manipulation and sorting techniques in the C programming language.”

(k) Optimization of Code
Homework 10:
(i) Optimize the following program.
(ii) Ask ChatGPT to optimize the same program.
(iii) Compare your answer with the one generated by ChatGPT. Which program is more readable and efficient?

Question to ChatGPT:
Optimize and make the following program more readable.

```
#include <stdio.h>
int main()
{ int i=21;
if(i==10)
printf("%d ", i);
else
if(i==15)
printf("%d ", 2*i);
else
if(i==20)
printf("%d ",3* i);
else
if(i==21)
printf("%d ",4* i);
else
printf("%d ", 5*i);
   return 0;}
```

Answer from ChatGPT (without editing):

“Here's the optimized version of the program:

```
switch(i)
{case 10:
printf(“%d=”,i);
break;
case 15:
printf(“%d=”,2*i);
break;
case 20:
printf(“%d=”,3*i);
break;
case 21:
printf(“%d=”,4*i);
break;
default:
printf(“%d=”,5*i);
}
```

This makes the code more readable and helps organize the logic based on the different values of i.”

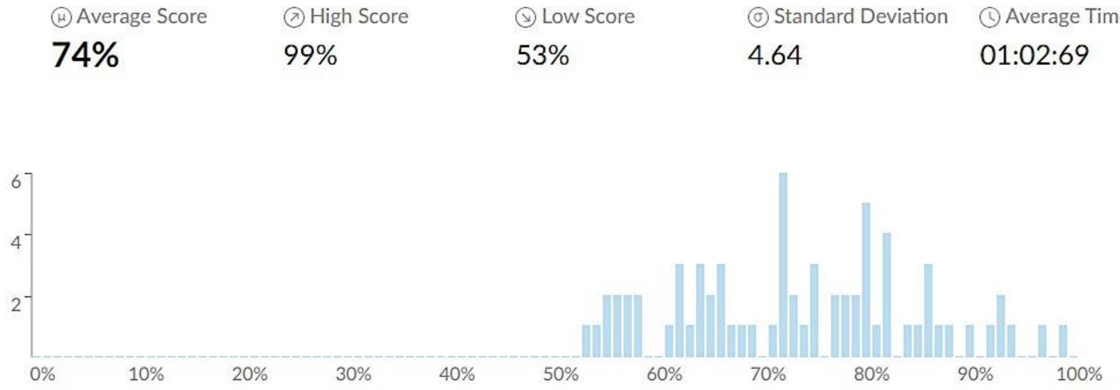
As you can see, ChatGPT used ***switch*** **s**tatement rather than multiple ***if/else*** statements, which is a better solution.

(3) Analyzing the collected data

The following data are collected from final exams in C++ programming course for three consecutive academic quarters. Students are undergraduates with engineering majors. Average sample size is 55.

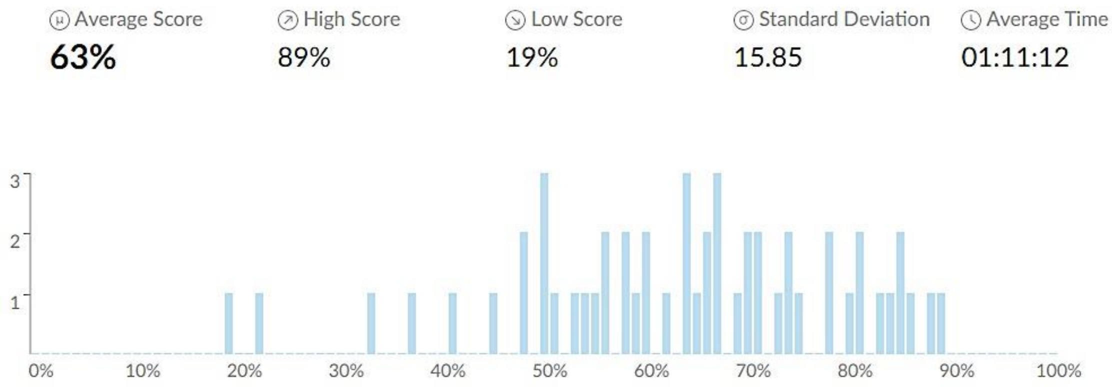
The results of exams for students learning from the teacher, as well as ChatGPT will be considered. Figures 1, 2, and 3 display the grade distribution for final exams in the Fall 2023 with ChatGPT and in Fall 2021 and 2022 without the use of ChatGPT. The results indicate the use of ChatGPT improved Average, High, and Low scores. Also, the use of ChatGPT reduced the standard deviation and grades are less spread out. Overall, the number of students who failed the class reduced due to use of this AI tool.

In Figure 1, average scores of exams improved notably with the use of ChatGPT. Average score improved to 74% compared to 71% (Figure 2, Fall 2022) and 68% (Figure 3, Fall 2021). Also in Figure 1, maximum scores of exams improved to 99% compared to 89% (Figure 2, Fall 2022) and 95% (Figure 3, Fall 2021). The number of students who failed the class reduced.



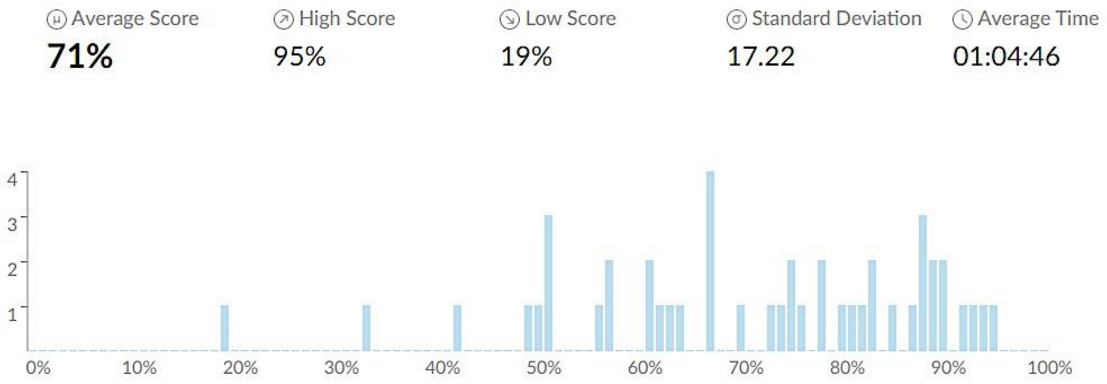
Average Score 74%
High Score 99%
Low Score 53%
Standard Deviation 4.64
Average Time 01:02:69

Figure 1: Final Exam in Fall 2023 (with ChatGPT)



Average Score 63%
High Score 89%
Low Score 19%
Standard Deviation 15.85
Average Time 01:11:12

Figure 2: Final Exam in Fall 2022 (without ChatGPT)



Average Score 71%
High Score 95%
Low Score 19%
Standard Deviation 17.22
Average Time 01:04:46

Figure 3: Final Exam in Fall 2021 (without ChatGPT)

Academic Quarter	**#Students/Quarter**	**Average Exam Score**	**High score**	**Low score**	**Standard Deviation**
Fall 2023	73	74	99	53	4.64
Fall 2022	54	63	89	19	15.85
Fall 2021	47	71	95	19	17.22

Table 1: Overview of Exam Scores

Table 1 compares the results across three quarters (Fall 2021, 2022, and 2023). Exam questions were the same in all three quarters.

(4) Conclusion

This paper demonstrates the use of ChatGPT. This tool has a wide range of capabilities in programming courses. It can help students to write a program and optimize codes with documents. The capabilities of ChatGPT provide explanations, examples, and its guidance makes it a useful resource. Teachers could save more time and cover other materials by referring students to AI tools for several concepts. In this paper, homework assignment was modified to increase the teaching value by ChatGPT and an instructor. Several homework assignments that could be used with Chat GPT were discussed. The data collected from a freshman programming course was analyzed with and without the use of AI. From derived data, students had a better understanding of the C++ programming language if AI tools are used. Based on the outcomes of the collected data, it is highly recommended to use AI for a student's first introductory programming course.

The main takeaway of the author from teaching this course with AI tool is that curiosity promotes learning. Overall, based on the author's experiences, most students were curious to learn more about this new tool. As a result, they were more engaged in writing codes. The author has warned students that scientific advances can both help and hurt and AI is no exception. Students highly welcomed this new tool and were eager to use it in a positive way in the other courses to enhance their learning.

(5) References

1. Kernighan, B.W., Ritchie, D.M.: The C Programming Language. Prentice Hall Software Series (1988).
2. Prechelt, L.:An empirical comparison of seven programming languages. IEEE Computer **33** (2000) pp. 23–29.
3. Sam Key, Python Programming in A Day & C++ Programming Professional Made Easy (Volume 43), Create Space Independent Publishing, 2015, ISBN-13:9781511776752.
4. Charniak, Eugene, et al. *Artificial intelligence programming*. Psychology Press, 2014.
5. Winston, Patrick Henry. *Artificial intelligence*. 3rd ed. Addison-Wesley Longman Publishing Co., Inc., 1984.
6. Phillips, Tanner, et al. "Exploring the use of GPT-3 as a tool for evaluating text-based collaborative discourse." *Companion Proceedings 12th intl. Conf. on learning Analytics & Knowledge* (2022): LAK 22.

7. Lund, Brady, and Wang Ting. "Chatting about ChatGPT: How May AI and GPT Impact Academia and Libraries?" *Library Hi tech news* (2023).