|  |
| --- |
| Cute yellow robot |
| AI Written Code Assessment  A1\_T2 |
| |  |  | | --- | --- | | Loai Hataba 20230553  Abdullah Mohammed 2023231  Hossam Abdelaziz 20230121 | CS213-OOP | |  |  | |

**AI-Written Code Assessment**

**Introduction**

* **AI Models Used**: Chat-GPT (4o mini) / Claude (3.5 Sonnet)
* **Initial Prompt**: Both Ai models were prompted with the same prompt

***“You’re a senior programmer with a lot of knowledge in C++, I want you to complete this class … (provide the header file)”***

One of the most important aspects of a good prompt is the persona, where you tell the ai model who it should be so it has a clearer path of where to search and which information to provide; so by telling the models that they’re programmers who are knowledgeable in C++ that would potentially help the results be better than just asking them to complete the class.

* **Modifications/Reprompting**: Both Models have been prompted more than once (with nearly the same prompts) to get somewhat of an acceptable result that matches the desired outcome.

**1. Correctness**

* **Chat-GPT:** 
  + Overall, the code was acceptable, the code works well in most normal cases in most operations, and the code seems to cover a lot of edge cases.
  + Yet when met with cases such as polynomials with complex it fails.
  + The get root function was a complete disaster even after a couple of tries to correct its code the model completely ruined the code making it unusable.
  + **Error Handling**: The model didn’t add any type of input checking or safeguarding the code, whenever an invalid input the program either runs forever and needs to be force closed or the there’s some kind of segmentation error or logical that forces the program to completely quit
  + The model could sometimes slip up and forget very basic and easy stuff that would seem not that important but every detail matters, and sometimes it ignores some of your request
    - Example: when provided the model to make the get root value give back multiple roots if exists and set default values for the parameters it failed to comply with the requests and yielded the code nearly as it is, even a bit worse, not adding the multiple roots feature nor setting the default parameters and removing the input asking the user for the guess and tolerance values, it had to be reminded again that it didn’t do what it was asked to do.
  + **Rating: 70%**
* **Claude:**
  + The code correctly implements polynomial operations and provides a comprehensive set of features for manipulation and evaluation
  + As for edge cases, the code removes leading zeros when constructing polynomials, ensuring that the polynomial degree and operations don't get skewed by unnecessary terms. However, input validation for edge cases like empty coefficients and division by zero in Newton’s method should be handled more thoroughly.
  + Most functions handled the tests well, except for the getRoot function, which struggled. This is reasonable, as it uses Newton’s method, an older technique that becomes less accurate as the polynomial's degree increases.
  + **Error Handling**: The code lacks robust error handling in some cases (e.g., division by zero in Newton's method, input validation). Adding checks would improve reliability, especially for edge cases.
  + While the model occasionally forgets to complete certain tasks, , it has never ignored a prompt entirely
    - For example, when asked to create a menu for operations, it only generated one for 8 operations and left out the remaining ones. Additionally, when prompted to edit the behavior of a specific function—such as its nature or return type—the model successfully made the changes to the function itself but neglected to update the menu or the header file. As a result, I frequently had to remind it to synchronize the menu with the latest function updates.
  + **Rating: 85%**

**2. Efficiency**

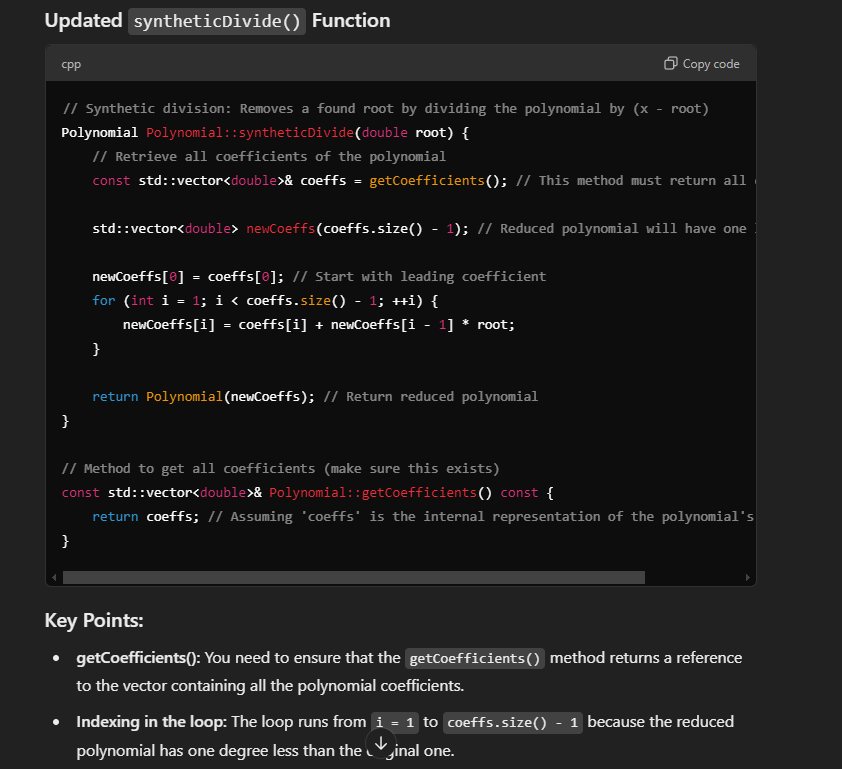
* **Time Complexity**: Both of the models had O(n) time in most of the methods averaging for about 4-8 microseconds (using chrono library) with the longest being O(n\*m) where there’s two inputs
* **Space Complexity**: Again, both of them used minimal storage where they just used variables and vectors.
* Both Models achieved very well in terms of speed and resource allocation in the best ways possible
* **Rating: 100%** (Both)

**3. Elegance**

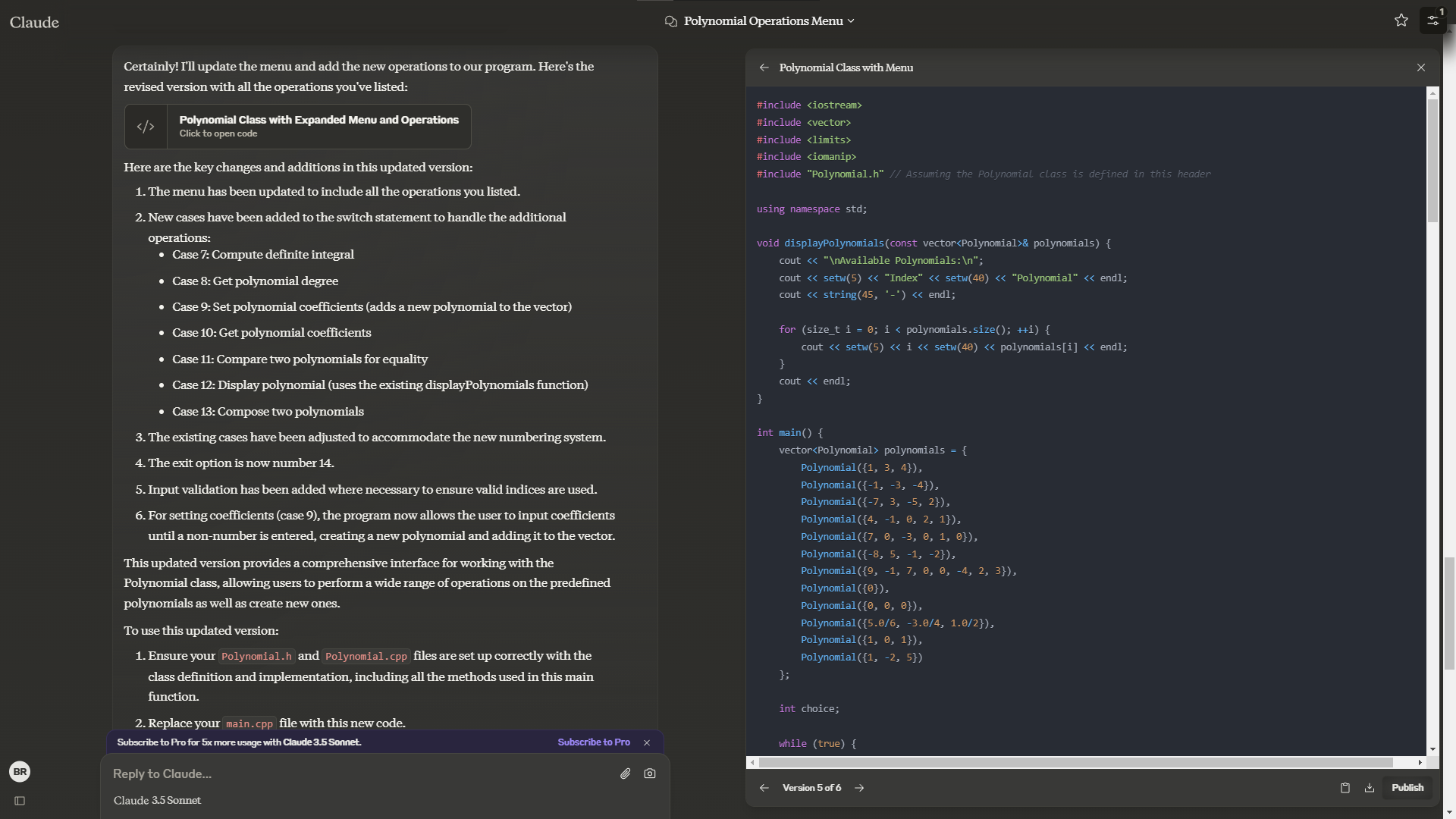
* **Chat-GPT:**
* Code elegance is definitely one of gpt’s strengths where using its huge training sets it figures out the best most elegant solution for a problem.
* The code overall didn’t take up a lot of coding space with a total of 470 lines of code (very acceptable).
* The model uses brilliant and efficient algorithms and techniques to solve problems and the tasks in as little coding space as it can possibly can, it achieves its goal with minimal, clear lines of code**.**
* **Rating: 85%**
* **Claude:**
  + The code balances functionality and simplicity. While operations like differentiation and integration are user-friendly, Newton's root-finding method can be somewhat overwhelming.
  + Claude efficiently managed the code space, completing the task in just 356 lines of code.
  + **Rating: 85%**

**4. Cleanliness**

* **Chat-GPT:**
* Despite the class being relatively not small, GPT wrote most code in clean and very human readable way with proper variable naming conventions and indentations.
* The model uses comments quite on point where it knows where a comment would be necessary to explain that line of code.
* A plus would be the color coding and code snapshots that the model uses to present the code, and also providing very detailed explanations under the code for what every part of the code does.



* **Rating: 80%**
* **Claude:**
  + **Code structure**: The code is structured well with clear separation of header and implementation. It follows common C++ practices with proper use of encapsulation and operator overloading.
  + **Naming conventions**: Most function and variable names are descriptive (e.g., degree(), evaluate(),derivative()), making the code easier to follow.
  + **Code comments**: The code could benefit from more comments, especially for more complex parts like the composition, root-finding, and evaluation algorithms. This would make it easier for future readers to understand the reasoning behind certain design choices.
  + Additionally, it is important to note that Claude is designed for complex tasks, such as code generation, which is reflected in its user interface by separating the code from its explanations.

****

* + **Rating: 70%**

**5. Summary:**

* **Chat-GPT:** 
  + Overall, the model did a fairly decent job in completing roughly the main idea of having a polynomial class with several methods
  + Yet aside from simple subtraction, addition, and simple integration the code is vulnerable to bugs and total program shutdowns
  + The experience using the model and having a conversation with it was definitely an easy task, perhaps a bit frustrating when you’re literally asking the model for something and it completely ignores it and responds with “of course here’s your updated code” and the code isn’t changed a single bit; another point is when the model gives you a fix for a problem but it completely obliterates another part of the code, where sometimes maybe the model can fix it or it maybe can’t and then you have no choice but to pray that you had some backup for your previous code.
  + We would give the model an overall rating of **75%**
* **Claude:**
  + **As for the code:**
    - The implementation features a well-encapsulated, object-oriented design. It effectively handles most edge cases, particularly concerning polynomial degree and coefficients.
    - there is room for improvement in error handling, especially related to root-finding and user input. Additionally, increasing the number of comments and documentation would help clarify the more complex sections of the code.
  + **As for the model:**
    - During my interaction with the model, the only Aggravating aspect was when it made edits to a function's code without synchronizing those changes in the menu or header. Aside from this issue, the model demonstrated a strong understanding of the prompts and provided efficient responses. Overall, it was a great experience.
  + Would give an overall rating of: 80%

**Links:**

-Chat-GPT conversation 1(class implementation): <https://chatgpt.com/share/670fd569-b568-8011-8501-c96948df8a49>

-Chat-GPT conversation 2(menu): <https://chatgpt.com/share/670fd5cd-ee64-8011-98e0-78e1dd90b984>

-Claude conversation 1: <https://claude.ai/chat/9eb78b11-cfd2-475a-8ed1-b93ae5c3155e>

-Claude conversation 2: <https://claude.ai/chat/70a67b7d-6719-4ba7-b29a-60b2665016fa>

-Claude conversation 3: <https://claude.ai/chat/4ddd0f76-8837-4db3-b5bc-3b30d3c1d2e0>

-Claude conversation 4(get root fix): <https://claude.ai/chat/80922f3e-7e1c-4e0b-8654-d6357522f834>