|  |
| --- |
| 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 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 it fails.
  + **!!!**
  + **Error Handling**: Assess whether the code handles possible errors gracefully (e.g., invalid inputs, exceptions).!!!!!
  + 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
* **Claude:**
  + **!!**
  + **!!!**
  + **!!!!**
  + Error Handling: Assess whether the code handles possible errors gracefully (e.g., invalid inputs, exceptions).

**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.
* **Optimization**: Are there unnecessary operations or steps that could be improved?

**Example/Observations:**

* (Provide information on the time/space complexity and potential optimizations.)

**3. Elegance**

**Overview:**

* How clear and simple is the logic of the code?

**Key Considerations:**

* **Conciseness**: Is the code unnecessarily verbose, or does it achieve its goal with minimal, clear lines of code?
* **Algorithm Choice**: Has the AI chosen an efficient and appropriate algorithm for the task?
* **Structure**: Does the flow of the program make sense and is it easy to follow?

**Example/Observations:**

* (Highlight areas where the code could be made more elegant, such as refactoring overly complex blocks into simpler ones.)

**4. Cleanliness**

**Overview:**

* Is the code easy to read and maintain?

**Key Considerations:**

* **Readability**: Assess variable names, comments, and structure for clarity.
* **Comments**: Are there sufficient comments to explain non-trivial parts of the code?
* **Formatting**: Is the code properly formatted with consistent indentation, spacing, and naming conventions?
* **Modularity**: Does the code follow principles like DRY (Don't Repeat Yourself) and modularity, separating concerns into functions or classes?

**Example/Observations:**

* (Note specific areas where the code is hard to read, poorly commented, or not modular enough.)

**5. Overall Recommendations**

**Summary:**

* Provide an overall evaluation of the AI-written code, balancing all the above aspects.

**Suggestions for Improvement:**

* Highlight areas for refinement and potential refactoring opportunities.

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>