

OOP and Design Patterns (CSCI 375)
Student Showcase (Final Project) Rubric

1. Project Title: Connect 4 with ChatGPT API
2. Team Members: Ben Loveland, Dillon McDermott, Liam Maddox
3. Evaluator: Dillon McDermott

Grading Rubric:

Instructions:

1. There are 9 technical requirements to grade the project and the team presentation.
2. For each requirement, use 0 - 5 scale in the Score column (0 - F, 1 - Needs improvement, 2 - Poor, 3 - Fair, 4 – Good, 5 - Excellent)
3. Use the *Notes* section to jot down any observations that may help in grading and justification.

Team and Technical Project Requirement	Score
<p>1. Use of fundamental OOD concepts, e.g.: Inheritance, Abstraction, Attributes, Getters, Setters, Methods, Modularity, Overloading, etc.</p> <p>Notes: We made sure to stay consistent with attributes, use getter and setter methods and have different classes that interact with each other demonstrating our use of Object Oriented concepts</p>	5/5
<p>2. Use of at least 3 Design Patterns -- presentation clearly stated and briefly explained design patterns use. Common design patterns are Iterator, Decorator, Observer, Strategy, Command, State, Singleton, Adapter, Façade, Flyweight, Abstract Factory, Composite, Template, MVC, etc.</p> <p>Notes: We used the design pattern's Factory, Observer, and State patterns as explained in our 4+1 views document. I will dock 1 point because I feel we could have had a more thorough explanation during the presentation itself though rather than just list them as we did.</p>	4/5

<p>3. Testing for correctness – automatically generates test data using hypothesis, usage of mocking/patching, provides code coverage and Python type check (mypy) reports, etc.</p> <p>Notes: Our program passed the mypy and flake8 test's and we received a high coverage with out unit testing that uses mocking, patching, and hypothesis.</p>	<p>5/5</p>
<p>4. Documentation – clear, easy to follow documentation, UML diagrams are complete, and notations are correct; explanation of objects interaction is clear and complete.</p> <p>Notes: We have multiple UML diagrams with both a complete program overview and diagram's for individual classes. We also have detailed doc strings for every method, class, and function and html documentation in the docs folder.</p>	<p>5/5</p>
<p>5. Software management – good usage of management, communication and tracking tools e.g., Gant chart, Kanban board, GitHub Project, Clickup, Discord, Slack, etc.</p> <p>Notes: Our group maintained consistent communication so we all remained on the same page and out project was well organized in GitHub. We didn't use issues or branches as much as we should have we still were effectively coordinate.</p>	<p>3/5</p>
<p>6. Teamwork – clear division of labor and progress tracking; helping each other, etc.</p> <p>Notes: Our work was well divided with each group member playing roles and assisting when needed. Ben handled the api implantation, I handled testing and started the server, Liam handled documentation, and we all worked together on whatever was needed with remaining topics so everyone pulled there weight.</p>	<p>5/5</p>

<p>7. Project requirements and execution -- clearly stated functional and technical requirements, project adequately challenging for sophomore-junior students; project demo was clear and concise, etc.</p> <p>Notes: Our project was clear to demonstrate and explain and was an appropriate level of challenge using multiple principles we learned in class as well as some topics not covered such as creating a server with html, css, and javascript.</p>	5/5
<p>8. Team presentation -- all members participated in presentation, used the visual and oral presentation techniques and tools to engage audience, etc.</p> <p>Notes: Our presentation primarily consisted of visuals such as our demon and diagram's, we covered all our bases such as documentation, testing, and our design, and every group member was present and participated.</p>	5/5
<p>9. Use 4+1 Views to explain the design to the audience.</p> <p>Notes: We have a well organized and detailed 4 + 1 view for our project effectively explaining our design, tools, and patterns. We also have a views.puml file and UML files to provide alternative options.</p>	5/5
<p>10. BONUS: Above and beyond – Team went beyond the above list e.g., great User Interface, use of Database, real-world application, client delight and interaction, CI/CD, deployment, etc.</p> <p>Notes: I feel our group were above and beyond in using a project that combined multiple options such as the use of a web based interface (python local server), machine learning application (chatGPT), and a simple game (connect 4). While also making the came look presentable and easy to use for any client. The web server in particular added to the project since the frontend and backend techniques weren't something covered in this class.</p>	5/10
<p>Total Score</p> <p>Note: Max score can be 55 due to 10 BONUS points.</p>	47/45