Coding Standards:

1. **Formatting:**
   - Follow Google C++Style Guide for code formatting
2. **Naming Conventions:**
   - Use descriptive and meaningful names for variables, functions, and classes
   - Follow camelCase for variable and function names
3. **Comments:**
   - Add comments to explain complex or non-obvious sections of the code
   - Keep comments concise and up-to-date
4. **Classes and Functions:**
   - Keep classes and functions focused on a single responsibility
   - Use encapsulation to hide implementation details
5. **Error Handling:**
   - Implement appropriate error-handling mechanisms
   - Provide informative error messages.
6. **Consistency:**
   - Maintain consistency in coding style throughout the project.

Contribution Guidelines:

How to Contribute:

1. Fork the repository
2. Clone the repository
   - Git clone https://github.com/lariosjulianna/blackjack.git
3. Create a branch
   - Git checkout – b feature-branch
4. Make changes
5. Run tests
6. Commit changes with clear and concise commit message
   - Git commit -m “Add feature: description”
7. Push changes
   - Git push origin feature-branch
8. Create Pull Request
   - Open a pull request from your fork to the main repository
   - Provide a detailed description of your changes
   - Expect and address code reviews

9. Submitting a pull request will trigger a review process, every change is reviewed before being merged into the main branch.
10. Update the README file if changes impact functionality or how tests run.

Setup and Testing Instructions:

Setup:

1. Prerequisites:
   - Ensure you have a C++ compiler installed (e.g., GCC).
   - Install any dependencies mentioned in the README file.
2. Clone Repository:
   - Clone the project repository to your local machine.
   - git clone https://github.com/main-repo/blackjack.git
3. Build the Project:
   - Use the provided build system or build scripts to compile the project.
   - make
4. Testing:
   - Run Unit Tests:
   - Execute unit tests to ensure the core functionality is working.
   - make test
   - Ensure new tests do not break existing tests. Run locally before committing and pushing.
5. Test with Sample Data:
   - Utilize sample data or create test scenarios to validate different game conditions.
6. Code Coverage:
   - Check code coverage using tools like gcov to ensure comprehensive testing.
7. Report Issues:
   - Report any issues or bugs encountered during testing.

Pull Request Process:

Review Process:

- Every pull request triggers a review process.
- Direct pushes to the main branch are not allowed.