

Final Project: Chess Tournament

Prompt:

The University of Houston Chess Club is organizing their first tournament! Students from all over the state are expected to join the tournament and battle for the title of chess champion. The Chess Club has tasked you with the design and deployment of a program to help them manage the games and keep track of the results.

Design Requirements:

Implement a Player class

- Create a class for managing information about the players. Each player should have recorded, at a minimum:
 - First, middle, and last name
 - Club or school of origin
 - USCF rating (0 – 2700, or unrated)

Implement the Tournament

- Provide a system for managing the tournament. The system should be able to perform the following tasks:
 - Enter players (manually or from file)
 - Create initial match pairings based on rating (the criteria are flexible, but we are trying to have balanced games between players of similar USCF ratings)
 - Generate subsequent matches based on the results of the first rounds.
 - Keep track of all games results, save and update the tournament state.
 - Keep track of the players' wins/losses, ranking, and determine the winner.

The tournament should include a first bracket round in which all players in a bracket play each other twice (once with the black pieces, once with the white pieces). The best in the bracket advances to the direct elimination round, which determines the final winner. You must be able to manage a variable number of total players in the tournament (dynamic allocation). The possible outcomes of a given game are win, lose or draw. You are allowed some flexibility in the design of the tournament rules, but you must explain your design choices.

System Requirements:

- Must use file I/O for saving and loading players' data and tournament progress between different sessions. The tournament results and ranking should also be saved to a different file.
- Must use at least three (3) classes (although more are probably needed) to manage the players and tournament.
- Allow for user input to update the progress of the tournament and file input to reload the status over several days of play.
- Use comments in your program to explain what each class/section of code does.
- Use text-styling (ASCII) to create an aesthetic menu system.
- Update the player's rating after each game using the USCF formula:
<http://math.bu.edu/people/mg/ratings/approx/approx.html>

Final Project: Chess Tournament

Submission:

1. **Prepare a Preliminary Project Report, due Tuesday, November 22nd.** This report should include a description of how the program is (or will be) designed, including the design of the classes and their inheritance relationships. Be sure to describe the main functionality (in the form of pseudo code or step-by-step instructions) of the program. Please also provide the menu options and their use cases for your program. You can refer to ZyBooks 11.10 to learn how to use UML to describe your OOP design.
2. **Prepare a short two to three-page report** explaining how the program was designed and developed, what classes were used, any special instructions to operate the program, and how the final implementation differed from the plan. **Submit the report on Blackboard by Friday, December 2nd.**
3. **Submit your source code as a single .zip file on Blackboard by Friday, December 2nd.**
4. **Provide a link to a video clip (less than 5 minutes in length) demonstrating all the above required functionality. Submit this to Blackboard by Friday, December 2nd.**

Grading:

Preliminary Project Report - **100 points.**

Peer-grading (based on the uploaded video) - **100 points.**

Final product graded by instructors and TAs - **200 points.**

Final report - **100 points**

Demo via video clip – **50 points**

Source code – **50 points**