Programming in Java
Prep Term 2018-19
Lab 1

P1. We would like to simulate a knock-out style tournament such as Wimbledon or the final rounds of the World Cup. The Tournament starts with *n* Teams (n is a power of 2). Each team is assigned a rank (or seeding) based on prior performance. The teams are arranged in some order (say, alphabetical or based on a random draw). The tournament proceeds in Rounds, where in each round, Matches are played by pairs of teams (based on the original order). The winner of each match advances to the next round, where again successive pairs of teams (winners from the previous round) play matches.

Write a Java program to model this, and to run through the matches and identify the winner of the tournament. The program should print out the results of each match, round by round. We would like the results of each match and round to be retained for subsequent analysis. What would be the class design and functionality of each class. You can use any rule to decide the winner of a match - e.g. the one with the better rank, or a random choice. You can use any mechanism to name the teams and assign rankings. You can assume that n is not very large.

Variant:
Assume that we want to try out many different rules for deciding the winner of a given match. Not all of these may be known when the program is first designed. However, you would like to not have to change too much of the code every time a new rule is implemented. How would you change the design of classes to allow this kind of “scalability”? Is it possible to design the code such that there are no changes to any of the existing classes except the “main” method?

P2. (For those who are proficient in Java and OOP - and general programming - and want a different challenge!)
Implement a set of classes for managing JSON objects. The classes should allow the querying and editing of JSON objects, as well as creating JSON objects by parsing a text file containing the JSON string.
For simplicity, you can assume that a JSON object consists of one or more key-value pairs (with the syntax *<key : value>*, separated by commas, where *key* is a string and *value* is one of

- A string
- An array of JSON objects
- A JSON object

Your program should be able to read in file containing a JSON object, and allow the parsed object to be queried. (You can print out the resulting object in any convenient way).

Can this be implemented as a one-pass parser where the input text/string is read only once.