For this course you will be required to write a program capable of playing the game Hop Step. The game will be described in a separate document and played together in class. This document outlines the milestones on the course project. All work on the project is to be your own. You are free to discuss the project and enhancements to your program with other students, but you may not share source code. List any students that you discussed each milestone with on your project report.

Part 1: Practice using server & turning in using SVN
Due January 11, 2011 at 12 noon
Software will be provided that plays rock-paper-scissors. Modify the program provided or write your own program that can play on the server. All programs, including several default programs, will be played against each other on the server. A bonus will be given to the student that writes the winning program.

Part 2: Implement Hop Step
Due January 18, 2011 at 12 noon
Implement the game Hop Step so that you can get a list of legal moves, apply moves, undo moves, and test for the end of the game. Write a short report providing information about the game gathered from random games. What is the average length of a game? What is the average branching factor throughout the game? Does the game seem to be biased towards the first or second player?

Part 3: Implement minimax and a basic evaluation function
Due January 25, 2011 at 12 noon
Implement the minimax algorithm and a basic evaluation function. Your evaluation function can still be quite simple (e.g., the difference in branching factor for each player). Write a short report describing your evaluation function. Play against the program on the server and report your performance.

Part 4: Implement alpha-beta pruning
Due February 1, 2011 at 12 noon
Add alpha-beta pruning to your implementation of minimax. Look at how move ordering influences the number of nodes expanded. Write a short report showing how you verified that your implementation of alpha-beta pruning was working and the reduction in nodes expanded with alpha-beta pruning. Estimate the reduced branching factor after applying alpha-beta pruning. Play against the program on the server and your previous program and report your performance.

Part 5: Implement iterative deepening
Due February 8, 2011 at 12 noon
Add iterative deepening and the history heuristic to your program. Write a short report detailing how the history heuristic improved the performance of your program. Play against the program on the server and your previous program and report your performance.

Part 6: Implement transposition tables
Due February 15, 2011 at 12 noon
Add transposition table to your program. Verify that they work correctly with alpha-beta pruning. (GHI is not a problem in Hop Step). Write a short report detailing the performance gains provided by
transposition tables, showing the difference in speed with and without transposition tables, as well as the number of nodes expanded. Play against the program on the server and your previous program and report your performance.

Part 7: Write the strongest program you can  
Due February 8, 2011 at 12 noon

Using any techniques that you would like, improve the strength of your Hop Step program. As you make each improvement to your program, compare the winning rate against your previous program and the program on the server. You may also want to play against other classmates to compare performance.