

Guidelines and Suggestions for Project Presentations, Fall 2005

Presentation guidelines.

1. Because we are doing presentations of projects over two periods for 14 people, assuming no teams, you have about 7 minutes per person. You may not exceed that by more than one minute, but you may end early. Teams may have 10 minutes.
2. No delaying your presentation except for emergencies. (Not including emergencies like: My demo doesn't work. Make sure you have at least a partly working demo.)
3. You are graded on the presentation as well as on the paper. It should be engaging; it should teach us something. Feel free to use visual aids: run a program, use PowerPoint, or supply a handout. (I'll fund copying if I have a handout a day ahead.) Anything to make sense of what you want to say! Don't get overly fancy and use visuals as a substitute for worthwhile material to present. Your tone should be professional, not familiar. Dress up on the day you present!
4. Hand me your paper and/or commented code at your presentation. For those whose presentation is the first day, your paper or code may not be complete, and that's OK. The written part of the project is not due for everyone until the end of the day December 9.

Guidelines for the **paper** are the same as for every lab. Don't just turn in a working program. Turn in a lab report in which you say: What is the problem to be solved, the solution, instructions for use of the program (if a program is involved), limitations, sample input and output, and conclusions.

Suggestions. Here are Lab 5 projects done by student in past years.

1. Computer worms.
 - See a sample worm at q:\InstructorFiles\Chase_Gene\Assembly\worm.s
2. Java simulator for a single-instruction computer
 - See textbook CD, page IMD2.20-10, and the folder q:\InstructorFiles\Chase_Gene\Assembly\singleInstructionComputer
3. Calculators
 - A simple 4-function calculators in reverse Polish notatin (RPN)
 - A scientific calculator (included square root and trig functions)
 - A 7-function calculator (percentages, factorial, ^, +-* / in floating point where possible and where necessary in integers, as in exponent and factorial).
4. Some games
 - Question game tree ("The animal game")
 - Trivia game with ranking system
 - Hangman
 - Memory game (where you remember what you've been shown earlier in the game)
 - A bit-oriented one-player version of Red Dog (or Acey Ducey)
 - Arena (a bit oriented game for decision-making invented by the student team)
5. Some things needing a pseudo-random number generator
 - The dice game Farcle (see Internet for a description of the game)
 - Higher/lower number guessing game
 - Blackjack
6. Some other assembly languages to write a program in and compare with MIPS
 - Intel 80386, • Motorola 68030, • SPARC, • TI Calculator assembly
7. Small compiler from a part of BASIC to MIPS
8. Grabbing output from keyboard directly, not buffering (See pages A-38 – A-40, §A.8 Input & Output.)
9. Various models of parts of the MIPS CPU
 - Macromedia Flash 16-bit carry-lookahead adder and 1-bit adder
 - 4-bit Register file from hardware [done by an engineer]
 - 1-bit alu in JavaScript
 - Demo of gate arrays (one used Java Swing; another demoed TkGate on Linux www.tkgate.org)