Project Rubric

R1. Function & Robustness (60%) threshold-progressive
R2. Design & Software Architecture (15%), threshold-curved
R3. Implementation Details (15%), threshold-curved
R4. Documentation (10%), threshold-curved
R5. Miscellaneous: not allocated for now. New items might be added if we find it necessary.

Note:
- Threshold-progressive: you will need to reach a minimal threshold of functionality for “pass”, typically 60%, and then additional points will be awarded based on the level of work accomplished. At discretion of the TA-instructor, a project that fails to reach the pass threshold may be given a second chance to resubmit, with reduced points.
- Threshold-curved: you will need to reach a minimal threshold of the stated criterion for “pass”. Then, grades of passing projects are curved.
- You cannot expect to get high marks on items 2-4 when you do not reach the “pass” functionality threshold.
- late submission penalty: 3% each working day.
- only materials submitted through CSNet turn-in will be graded. Once a project is graded, which can happen anytime after you submit it to the CSNet, your resubmission will no longer be recognized.
- materials submitted in incorrect formats need to be resubmitted (any no-penalty extension at discretion of TA)
- Milestone grades within a project are evenly distributed, unless explicitly adjusted by TA-instructor.

R1. Milestones and functionality (Measure Function & Robustness)
1. each project will be broken down into a set of milestones
2. milestones provide short term objectives and test functionality
3. each milestones builds on functionality of previous milestones
4. test input and output files will be provided
5. evaluation of milestones with separate input and output files

R2. Software Architecture (Design)
1. modularity & organization,
   a. example: diagrams, flow charts, etc to describe high level logic
2. describe interactions between modules:
   a. example: analysis of dependency between modules, I/O structures, etc
3. requirements:
   a. Example: Proof/justification of models or algorithms to meet the requirements

R3. Implementation Details (reference: slides)
1. commenting: Note the contributor of each module/function.
2. naming (files, classes, functions, variables, etc.)
3. style (spacing, structure, length of functions, etc.)
4. organization

R4. Documentation:
Can a trained professional not on your team understand your work? It needs to start from high level introduction of concepts, requirements, team member duties (must be rotated at different stages), individual contributions.
1. project activity log (export and filter your svn log to the report pages, plus meeting records)
2. in wiki/web format (link to your source codes)
3. well organized (to support scaling up of the documentation process later)
4. polished & complete