

# Low RISC-V



## Build Tools to Help Your Peers While Learning the hot new ISA, RISC-V



CS@Mines Faculty Prof. Amelia Read ([read@mines.edu](mailto:read@mines.edu)) has been at Mines since Fall 2020, teaching CSCI 341 Computer Organization. This is a required course in the CS curriculum, teaching students how the insides of a computer works by studying the Instruction Set Architecture (often considered to be assembly programming by students), the CPU Datapath, and the Memory Hierarchy. Be part of the future of CSCI 341!

Prof. Read is looking for a team to make key improvements to CSCI 341 Computer Organization:

- The course is being updated to the hot new ISA, RISC-V
- The programming portion of CSCI 341 will be enriched with automated, low-stakes programs to give students practice with assembly programming – this is where you come in!
  - The bartik autograder, which hosts APTs for CSCI 261 and 262, will be our user interface, so you will need to build an understanding of the existing framework and how to add new language support and new self-graded exercises to it.
  - RARS, the RISC-V version of MARS, will be used to assemble and run RISC-V code – this existing program should work as it exists with no further changes.
  - MUnit, a Java unit testing framework for MIPS, will be ported by the team to support RISC-V – we refer to this as RUnit.
  - The glue to use RUnit within APT will be developed by the team.
  - An initial set of low-stakes programs to write in RISC-V will be developed by the team.
- Stretch goal: develop program exercises for the assembly language modules of the course.

Students should be comfortable programming in Java, and familiar with MIPS programming – if you’ve completed CSCI 341 and have done Java programming, you’re ready! If you have Git, JUnit or testing experience, that’s a bonus! You will be learning RISC-V, how to test RISC-V code, how to port code from one ISA to another (MUnit->RUnit), and how to integrate systems (RUnit & APT).

Preferred team size: 3-5 – the more students that are involved, makes the stretch goal feasible from the start. Work can be done remotely or in Mines computer rooms – no need to head down a long road for this one!

MUnit, RARS, and the autograder are open source; work done on those codebases will be open source. Students need to assign ownership of intellectual property to the client for the programming exercises they develop.