

# Intermediate Milestone 4

Tao Wang  
CS 6460: Educational Technology  
Spring 2016

<b>Week #</b>	<b>Due Date</b>	<b>Milestone or Task</b>
13	4/10/16	Implement loading, editing, and compiling source files. Bug fixes and testing.
14	4/17/16	<i>Intermediate Milestone 4</i> Deliverables (1) Prototype satisfying #1-3 from project description. (#4 time permitting) (2) Documentation. (3) Outline of project paper.

## Overview

This is the fourth milestone of a project to develop a Java interpreter that can step through code segments and produce line-by-line code traces and visualizations. The motivation for this work is to provide a tool that can assist new computer science students in doing code traces, debugging, and problem solving. The tool can also be used by instructors in preparing example code walk throughs for lectures or tutors and teaching assistants when demonstrating code for students in one-on-one sessions.

## Previous Documents

- [Proposal REVISED](#)
- [Milestone 1](#)
- [Milestone 2](#)
- [Milestone 3](#)
- [Milestone 3 video demo](#)

## Progress

Requirements from the project proposal:

1. Must have: Display visualizations of program state: values of primitive variables, fields of objects, contents of arrays. ✓
2. Nice to have: Create, load, save, and edit Java source files locally. ✓
3. Nice to have: Compile and run Java programs. ✓

All the key requirements for this project have been met! A stretch goal of handling expression evaluation is going to be put aside in favor of more testing, improving the UI, and having more time to complete the paper and presentation.

## Code

Repository: <https://github.com/tao-wang/JavaCodeTrace>

The repository contains an Eclipse project with the current version of the project as well as this document and others.

## Paper Outline

Proposed conference: ACM Special Interest Group on Computer Science Education 2017 ([2016 website](#), [paper format](#)). SIGCSE 2016 has already come and gone, but I believe this conference would be the correct place to present this project.

- Motivation
  - Many beginning students have difficulty extrapolating past simple tasks
  - Programming languages are powerful but not user-friendly
  - How can students develop independent problem solving skills?
- Constructivism and CS Education
  - Students learn by creating models of knowledge
  - Students approach beginning CS with a very limited model of computing
  - The computer provides an "ontological reality", a formal framework to work within
  - It's important that tools represent that reality and make it explicit to students
- Tools to build models
  - A beginner's tool should provide an extra layer of representation
  - This representation can serve as a strong set of "training wheels" for new students
  - Tracing code is a skill that requires a correct mental model of computing
  - A code editor that assists in code traces would help students begin to think about code independently
- Design of Java Code Trace
  - Java interpreter
  - Snapshot format
  - Visual representation
- Use cases
  - Tracing assignment statements
  - Tracing loops and conditionals
  - Tracing method calls
  - Tracing recursive calls
- Discussion
  - Initial feedback
  - Improvements needed before distribution
  - Outline for formal study of effectiveness