

# Java Tutor: Bootstrapping with Python to Learn Java

Casey O'Brien, Max Goldman, Robert C. Miller

MIT CSAIL

Cambridge, MA 02138 USA

{cmobrien, maxg, rcm}@mit.edu

## ABSTRACT

A common pattern among undergraduate computer science curriculums is to teach an introductory subject in Python followed by a more advanced software engineering subject in Java. We are building an online tool that will help students who already know Python learn the syntax and semantics of Java. Our system will differ from existing online tutors and tools for learning Java in two main aspects. First, our tutor will focus on the transition from Python to Java. Using this basis will allow us to gloss over basic concepts of programming which students are already familiar with and focus on the specifics of Java. Second, our tutor will crowdsource writing test cases for problems to the learners themselves. This will give students practice writing tests, and will also reduce the burden on instructors, who would otherwise need to implement test suites for every problem in the tutor.

## INTRODUCTION

Students often learn how to program in a scripting language like Python, and then later take a more advanced software engineering class in another language, like Java. For students who have never programmed in the new language before, this can be a challenging transition.

MIT undergraduates studying computer science begin their careers by learning Python in 6.01 (Introduction to Electrical Engineering and Computer Science). With the help of an online Python tutor [?], students are able to learn the syntax and semantics of the language. A few semesters later the students enroll in 6.005 (Software Construction), where they are taught principles and techniques of software development. At the same time, they are expected to complete assignments demonstrating their understanding of these concepts in Java.

We are building a Java tutor which supports students in 6.005 learning Java by translating Python code to Java code. The main goal is to build a tutor which is useful for students and easy for instructors to use. The class has about 200 students each semester. Ultimately, we hope that this Java tutor will be used for 6.005x, the online version of 6.005 available through edX. This would expand the user base to many thousands.

The concept of learning one language by translating from another is not a novel idea. Consider a native English speaker who wants to learn Spanish. To do so, they would begin by learning how to translate words and phrases from English to Spanish, and then would build up to translating full sentences. No one would try to learn Spanish without taking advantage of their knowledge of English. Using the same reasoning, we hope to help students learn Java using Python.

The tutor will take the form of an interactive website, where students are presented with Python code and asked to write the corresponding Java code. A sample problem is shown in Figure 1. Their solutions are then checked against a JUnit test suite, and they have the opportunity to submit a new solution if test cases fail.

After creating a variety of sample questions, it has become clear that the process of creating questions will be time consuming for instructors. The focus on previous knowledge of Python helps decrease the instructor workload by creating an easy way for instructors to express problem statements. Instead of having to carefully write problem statements in English, instructors simply have to write out the code in Python.

Even with this advantage, writing all the necessary Python code and JUnit test suites would still be very time consuming. In order to further reduce the workload on instructors, our system requires students to contribute test cases to the test suite. Using student-written test cases, our tutor will create a comprehensive test suite which students solutions will be tested against.

## CURRENT RESOURCES

Like the online tools for learning Python (e.g., [?], [?]), there are many online tools for teaching Java through online exercises. Two examples are LearnJava [?] and CodingBat [?]. Both sites allow students to practice writing Java code by presenting problems and allowing students to check their solutions against test suites. However, neither site draws upon any other knowledge of programming languages.

Cody is a MATLAB Central game which aims to teach users how to write MATLAB code [?]. It allows players to interact with each other by creating new problems, viewing other solutions, and commenting on and liking problems and solutions. In this way, it takes advantage of input from all the users in the system to support learning.

## INSTRUCTOR WORKLOAD

In order to implement our tutor as described, each problem will require a brief problem statement, Python code, a Java template, a Java solution, and a JUnit test suite. As shown in Figure 1, the student will initially be presented with the problem statement, Python code, and Java template. The JUnit test suite will be written collectively by the students, and the Java solution will be used to check the accuracy of test cases.

After creating a variety of sample questions, it has become clear that the largest bottleneck in the process of creating problems is writing the JUnit test suite. In order to reduce

the workload on instructors, our system requires students to contribute test cases to the test suite.

Instead of developing a full test suite when creating a problem, the instructor will write a few very basic test cases. For a student attempting the problem, the area for inputting Java code will initially be locked. This will only be unlocked once the student has contributed a test case for the problem. Only once the student has written a test case which compiles and passes the instructor solution will the text area be unlocked.

Once a student submits a solution which compiles and passes their own test case, their solution will be run against the current test suite, and the student will be notified of the results and given the opportunity to fix their solution if any test cases fail. If another student later submits a test case which breaks a student's solution, that student will be notified and will have to go back and fix their solution.

In order to keep the size of the test suite from growing linearly with the number of students, de-duplication will be performed on the test cases. A new test case can be considered a duplicate of an existing test case if it both fails on the same set of student solutions as the other test case and executes the same lines of code. We will compute code coverage using the JaCoCo Java Code Coverage Library [?].

## SCALING

Currently, the tutor performs all compilation and execution server-side. For the initial user base of around 200 students in 6.005, we believe that this solution will be sufficient. However, server-side compilation will almost certainly be insufficient to handle the 6.005x user base of thousands of students.

One possible solution to this scaling problem is to perform some or all of the computation client-side. This could be achieved using a Java runtime and compiler implemented in Javascript. To do this we can use Doppio [?], a project which aims to implement an entire JVM in Javascript. The system could also combine the client- and server-side approaches, for example by compiling a student's submissions in the student's own browser, but running test case de-duplication on the server.

## PRELIMINARY RESULTS

The effectiveness of the tutor will be evaluated by measuring student performance and instructor workload, and by surveying both students and instructors. User testing on the full system will begin during the Spring 2014 semester. Initial tests on parts of the system offer promising results. Six students who were enrolled in either 6.005 or 6.01 at the time were asked to solve sample problems translating code from Python to Java. They reported that using Python as a basis was helpful in learning Java, and offered useful feedback.

To ensure that it is possible to write a comprehensive test suite from individuals each writing single test cases, we asked ten programmers comfortable with Java to contribute a single test case for the example in Figure 1. When we allowed users to view the current test suite, we found that the final test suite was comprehensive and also contained minimal duplication.

## CONCLUSION

We are building a Java tutor for Python programmers which is both useful for students and involves a manageable amount of work for instructors. The concentration on translation from Python to Java will allow students to take advantage of the knowledge they already have and learn Java more efficiently. The crowdsourcing of the test suites will allow instructors to easily create new problems, which will in turn provide additional practice for students.