

Preliminary Assessment of the Java Intelligent Tutoring System

Edward R. Sykes

*School of Applied Computing and
Engineering Sciences,
Sheridan College,
1430 Trafalgar Road, Oakville,
Ont., Canada, L6H 2L1
email: ed.sykes@sheridanc.on.ca
phone: (905) 845-9430
fax: (905) 815-4035*

Franya Franek

*Department of Computing and
Software, Faculty of Science,
McMaster University,
1280 Main Street W., Hamilton,
Ont., Canada, L8S 4L8
e-mail: franek@mcmaster.ca
phone: (905) 525-9140*

ABSTRACT

In an effort to support the growing trend of the Java programming language and to promote web-based personalized education, the Java Intelligent Tutoring System (JITS) was designed and developed. This tutoring system is unique in several ways. Most ITS require the teacher to author problems with corresponding solutions. JITS, on the other hand, requires the teacher to only supply the problem and problem specification. As a result, JITS intelligently examines the student's submitted code and determines appropriate feedback based on a number of factors such as the student's skill, JITS cognitive model of the student, and problem details. JITS is intended to be used by beginner programming students in their first year of College or University. This paper discusses the preliminary assessment of the Java Intelligent Tutoring System at the Sheridan Institute of Technology and Advanced Learning. The results of a qualitative investigation are presented describing the benefits and suggestions for improvement of the Java Intelligent Tutoring System.

Keywords: Web-Based Education, Evaluation of Programming Tutors, Intelligent Tutoring Systems.

1. INTRODUCTION

This paper presents the preliminary assessment of the Java Intelligent Tutoring System used by students and faculty. A brief overview of this ITS is presented, the qualitative methods used, and the results.

2. JITS USER INTERFACE

The user interface implemented in JITS is representative of fundamental features of a professional programming Integrated Development Environment (IDE). Students are presented with a problem, the problem specification, the skeleton code, the code editor, and a number of buttons with which to interact with the tutor. For example, once the student is ready to submit the code to JITS, the 'Submit' button may be pressed. The student can then view the hints by pressing "View Top Hint" or "View

All Hints". The hints are dynamically generated based on the problem details and the student's submission. At any time, the student can see their performance by pressing the 'My Performance' button the student is presented with a summary of his or her performance. Figure 1 depicts the user interface for JITS.

3. JITS ALGORITHMIC DESIGN

The Java Intelligent Tutoring System's intelligence is accomplished by an embedded logic module called JECA (Java Error Correction Algorithm). This module performs a number of operations behind the scenes. It implements a sophisticated scanner and parser that autocorrects the student's code when appropriate as well as generates a number of parse trees that have small permutations. This module then attempts to compile the best trees to ascertain the most likely path the student 'intended' to follow. With this knowledge, JITS can efficiently and effectively tutor the student [1].

The goals JECA are to:

- i) intelligently recognize the 'intent' of the student;
- ii) analyze the student's code submission;
- iii) 'auto-correct' where appropriate (e.g., converting "While" into the keyword "while", "forr" into "for", etc.);
- iv) learn individual student's misconceptions, and categorizes the types of errors he/she make;
- v) produce a 'modified code' that will compile (or bring the code closer to a state of successful compilation),
- vi) produce a 'modified code' that will meet the program specifications (or bring the code closer to meeting program specifications); and
- vii) prompt the student programmer for information when necessary via well-defined hint support structures.

JECA, combined with a well-defined student modeling mechanism and dynamic hint generation capabilities, enables JITS to significantly improve the performance of beginner Java programmers.

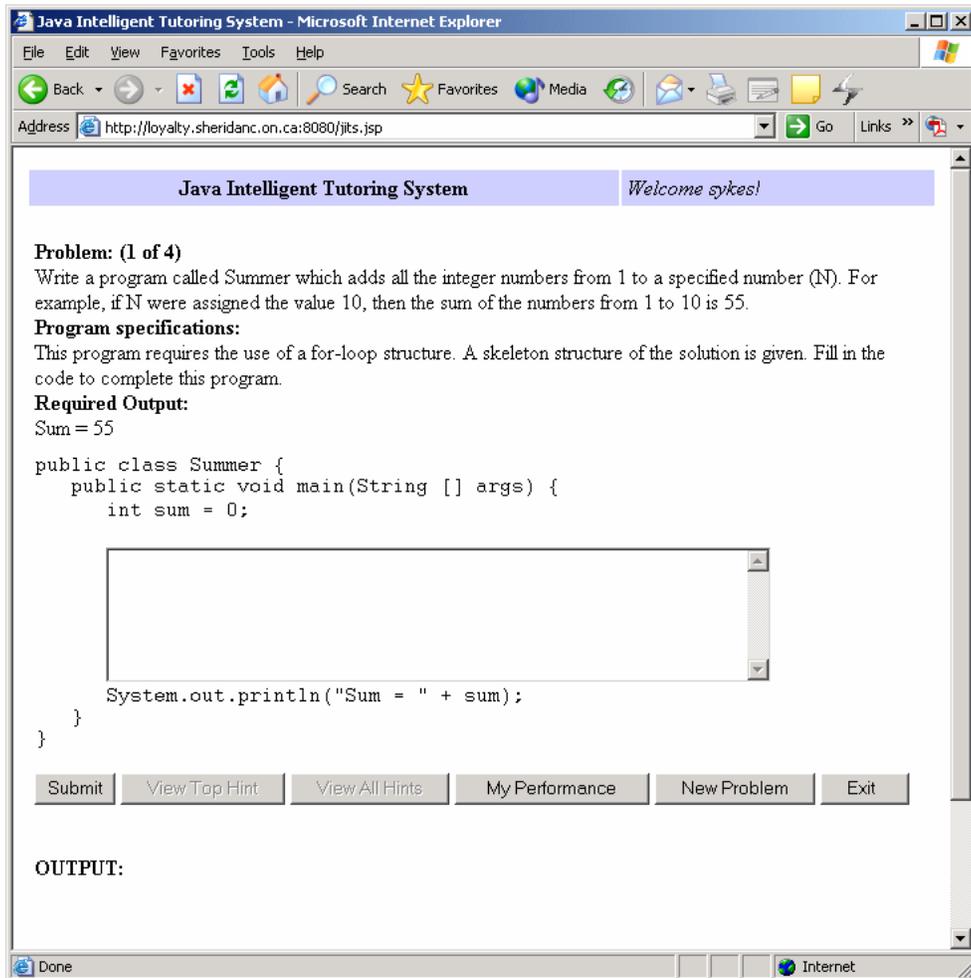


Figure 1. JITS User Interface.

4. JITS AUTHORIZING TOOL

An authoring tool is currently being developed which provides the teacher a convenient means to add problems to the database for JITS to use. This is a very easy process because the teacher only needs to provide the following information:

- the problem statement;
- the problem description;
- the required output; and
- the skeleton structure of the program.

As a result, the JITS authoring tool is intended to be extremely user-friendly and easy to add many problems of various levels of difficulty. Once the teacher has submitted the problems they are immediately available to JITS and thus students of the system. The Authoring Tool login screen is shown in Figure 2. Figure 3 depicts the Authoring Tool user interface.

The Authoring Tool provides a means to view the all the problems in the lesson set and edit selected problems [2]. In the Java Intelligent Tutoring System, the author of problems does not provide a solution. JITS carefully

scrutinizes the student's submission based on the problem description, specification, required output and template code is used by JITS uses to determine the appropriate feedback to the student. This ensures the greatest degree independent knowledge creation for each student [3], [4].

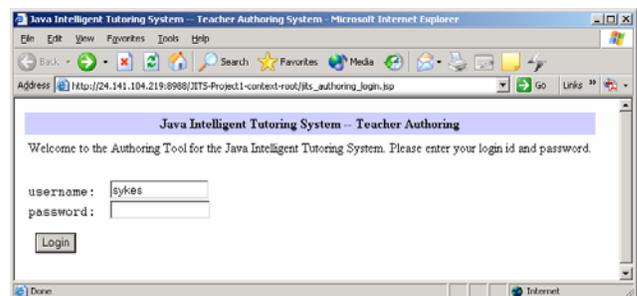


Figure 2. JITS Authoring Tool Login Screen

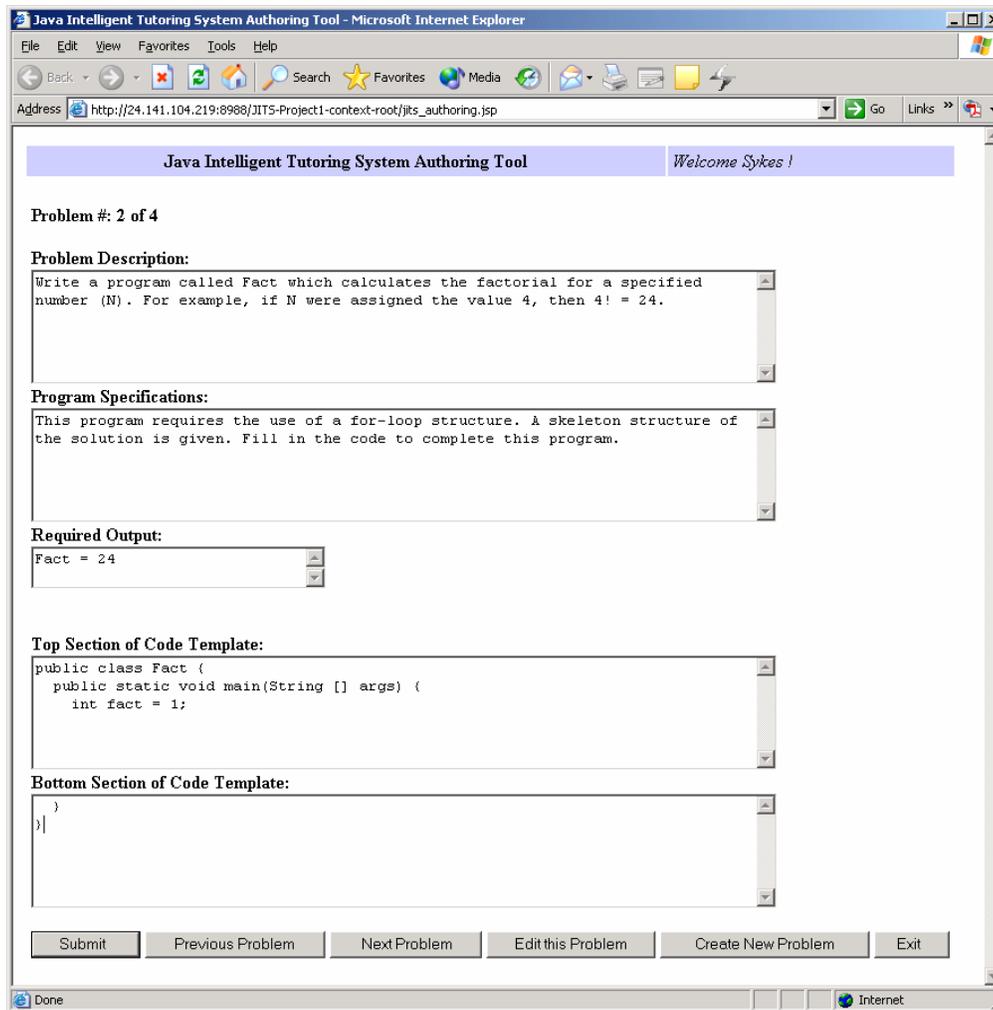


Figure 2. JITS Authoring Tool User Interface

5. QUALITATIVE INVESTIGATION STRATEGY

An interview-style survey sheet was constructed for this study. The survey includes 6 open-ended questions to facilitate a great number of perspectives and opinions. Table 1 presents the actual survey. By presenting the survey to students and teachers who have used JITS, feedback representative of these two perspectives was gathered. This information will be important to revise and improve JITS. The selection of students is described in the Sample and Population section.

6. SAMPLE AND POPULATION

The student sample base consisted of 350 students. From these, 6 students were randomly selected for a ½ to 1 hour long interview with the researcher to elicit specific information about their experience with the Java Intelligent Tutoring System.

The selection of Professors was based on a number of factors including their knowledge of the Java programming language, level of course offerings, and interest in offering critical opinions on the Java Intelligent Tutoring System. A total of 4 Professors were randomly selected for this qualitative study.

Table 1. Qualitative interview sheet.

Qualitative Project Interview					
<p>I am conducting a survey of those participants who were taught using the Java Intelligent Tutoring System at Sheridan. The information gathered from our interview will be used for my research. This involves determining the effectiveness of learning in this environment. For each question select the most appropriate response based on the following scale: 1 = strongly favorable to the concept, 2 = somewhat favorable to the concept, 3 = undecided, 4 = somewhat unfavorable to the concept, 5= strongly unfavorable to the concept. The following questions will be asked during the interview.</p>					
1.	How do you rate the Java Intelligent Tutoring Systems usefulness?				
	Very Useful			Not Useful	
	1	2	3	4	5
	Comments: _____				
2.	Do you feel the Java Intelligent Tutoring System is beneficial to your studies? List and explain the advantages/disadvantages of this learning environment.				
	Very Beneficial			No Benefits	
	1	2	3	4	5
	Comments: _____				
3.	Compare JITS with a traditional classroom. Do you feel JITS is better or worse than an ordinary classroom teaching environment? Identify any similarities and differences between a traditional classroom experience and the JITS learning experience.				
	JITS is much better than traditional classroom			JITS is much worse than traditional classroom	
	1	2	3	4	5
	Comments: _____				
4.	How do you rate the ease with which you use and understand the tutoring style of the JITS?				
	Very easy to use & understand			Very difficult to use & understand	
	1	2	3	4	5
	Comments: _____				
5.	Have you enjoyed JITS? Explain why or why not.				
	Very Enjoyable			Not enjoyable	
	1	2	3	4	5
	Comments: _____				
6.	Do you feel you learn more detailed information or about the same as a regular classroom when using JITS? Explain why or why not.				
	Learn Better			Learn the same	
	1	2	3	4	5
	Comments: _____				

7. FINDINGS

Overall, the students found the Java Intelligent Tutoring System enjoyable, beneficial, and useful. Table 2 depicts the summary statistics of the qualitative survey. It can be seen that JITS performed above average in all categories and scored the highest in two categories: “Usefulness”, and “Ease of Understanding Tutoring Style”. However, the findings did reveal important issues to improve JITS. The comments that follow will be reviewed for potential inclusion for later releases of JITS.

The Student’s Perspective

All of the students enjoyed working with JITS. Many voiced they are pleased with the following:

- 1) Feedback mechanism – It provides hints quickly and to the point. The hints are also not overwhelmingly complicated – quite unlike traditional compilers.

JITS Qualitative Summary Results – Students	
1. Usefulness.....	90%
2. Beneficial	75%
3. JITS is better than a traditional classroom.....	50%
4. Ease of JITS Tutoring Style.....	90%
5. Enjoyable.....	85%
6. Learn Better.....	75%

Table 2. JITS Qualitative Summary Results for Students

- 2) One student stated, “[JITS] tells me the exact spot in the code where I need make my correction – I like that. I wish other systems would do that.”
- 3) JITS helps students solve syntax and logic errors while developing a solution to a problem. One student stated, “I am definitely learning better in this environment than in a traditional environment.”
- 4) Integrated Development Environment – similar to professional programming environments.
- 5) Many students stated that they felt JITS was very useful since it is available 24X7 and all a student needs is a browser.
- 6) One student said, “Can we have this system in our course from now on?”

Regardless of the apparent success of JITS, there were suggestions from students to assist in making it even better. The following section include comments from this perspective.

The first suggestion is related to the navigation in the Java Intelligent Tutoring System. Currently, JITS is designed to encourage students to move forward through the problem sets. In its current form, there is no “previous problem” button. Some students stated that it would be convenient to be able to navigate forward and backward through the problems. This has two main benefits: it provides more freedom within the ITS and consequently the student has more control over his/her own learning, it also allows the student to select the problems they are more interested in solving and/or more able to solve. This view is similar to popular testing strategies such as Prometric Self-test software, and traditional test taking techniques [5], [2].

Even though JITS is designed with an integrated sequencer which selects problems that are a suitable level of difficulty and matches these problems with the student’s skill level it was felt that students should have the option to navigate and have more control over their learning process.

The second suggestion is related to the difficulty rating for each problem. Currently the problem scale has only three values: “Easy”, “Average”, and “Hard”. Some students felt that by introducing a wider range of values for the problem difficulty it would provide better sequencing for the student. As a result, JITS would be better able to select more appropriate questions for the student.

The third remark raised by the majority of the students was the “My Performance” button’s output. (Please see Figure 1 of the Java Intelligent Tutoring System.) A sample output after pressing the “My Performance” button is shown in Figure 3.

<p>OUTPUT: My Performance:</p> <p>Problems Attempted: 4 Problems Solved: 2 Overall Performance: 72</p>

Figure 3. Sample “My Performance” output.

Students felt that a more detailed representation of their performance could be helpful. Currently, JITS takes a number of factors into account when computing the student’s performance. JITS correlates the student’s skill level with the problem difficulty and ranks the points accordingly. JITS determines the number of problems solved against the number of problems attempted in combination. Furthermore, JITS combines this information with other gathered facts about the student as represented in the established student model. See Figure 4 for a depiction of the student model and related modules. As can be seen from the diagram, there is a tremendous amount of information available for analysis regarding the student’s performance. Future versions of JITS will provide much more detail regarding student’s performance.

Students felt that the hints were extremely good when the programming error was a syntax error. The Java Error Correction Algorithm (JECA), for the most part, was able to determine the intent of the student and offer meaningful and helpful corrective feedback. However, some students suggested better viewing of the syntax errors would be helpful. For instance, the use of a pointer:

```
for (int i=0; i<10 i++)
    ^
```

would be helpful for students to see that the corrected code should be:

```
for (int i=0; i<10 ; i++)
```

This feature is common among many language compilers (e.g., Fortran, Pascal, Turing, Java, etc.). With this “pointer”, the student is becoming more familiar with what they can expect to see in real programming environments.

Overall, however, the students seemed quite happy with the prototype of JITS. They all seem eager to see and try out future versions of the Java Intelligent Tutoring System.

The Professor’s Perspective

The section summarizes the views of Professors involved in this study. Table 3 shows the statistical results of the interviews.

JITS Qualitative Summary Results – Professors	
1. Usefulness.....	85%
2. Beneficial	80%
3. JITS is better than a traditional classroom.....	40%
4. Ease of JITS Tutoring Style.....	80%
5. Enjoyable.....	75%
6. Learn Better.....	75%

Table 3. JITS Qualitative Summary Results: Professors

Many Professors said they are pleased with JITS in the following ways:

- 1) One Professor stated, “The embedded logic unit called JECA is a sound tool – it picks out the most significant error the student need to focus on. I feel the student is developing core programming debugging skills with JITS.”
- 2) Integrated Development Environment – similar to professional programming environments.

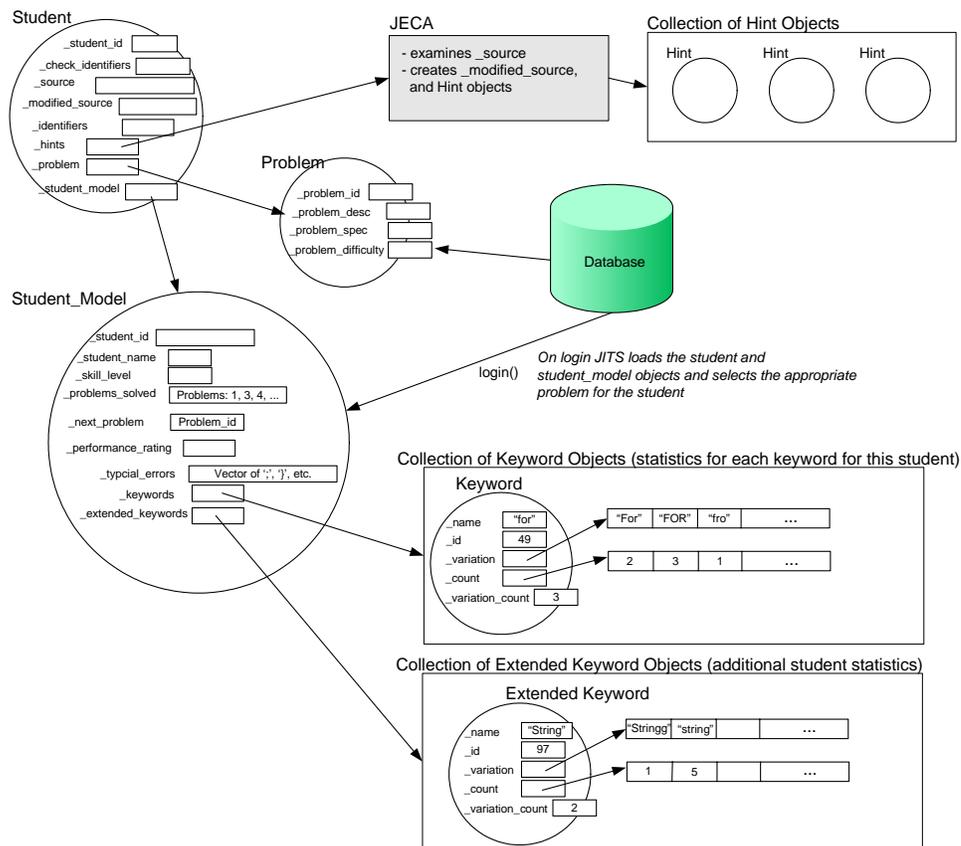


Figure 4. JITS student model and related modules.

- 3) Many Professors said that they would like to use JITS to augment their existing Java courses. They felt that JITS provides a means for students to receive extra tutoring when the Professor is not available.
- 4) One Professor said, "The quality of tutoring that JITS performs is comparable to a human tutor."
- 5) All of the Professors said that they liked the fact that there was no client installation required for them or their students.
- 6) Many Professors were happy that JITS was available 24X7. This makes it easier for students to work on problems at their own time and at their own pace.

One Professor suggested that JITS could produce a report representing the student's performance over a period of time. This would also be helpful to identify students who need additional assistance. It could also be used to identify those students who are doing extremely well and may be interested in more challenging problems.

All of the Professors enjoyed using the JITS Authoring tool. (Please see section 4 and Figure 2.) Although still under development, the prototype made Professors aware that they can easily create, edit, and review problems. Once the problems have been added they are immediately available to their students. A second benefit Professors stated was the fact that they needed only a browser to access the Authoring tool and JITS. Custom client installations are not required to use the Java Intelligent Tutoring System and the Authoring Tool. The majority of Professors in this

study felt that this 24X7 access from any Internet connection was a very good feature.

8. CONCLUSIONS

The Java Intelligent Tutoring System prototype has been a success. JITS was met with interest by students and Professors alike. After trying several problem sets both groups were happy with the performance of JITS. The various issues and suggestions raised by students and Professors are being reviewed. Integration of some of these requested features will be available in the future releases of the Java Intelligent Tutoring System.

9. REFERENCES

- [1] E. R. Sykes and F. Franek, "A Prototype for an Intelligent Tutoring System for Students Learning to Program in Java," *International Journal of Computers and Applications*, pp. In Press, 2004.
- [2] C. N. Rowe and P. T. Galvin, "An authoring system for intelligent procedural-skill tutors.," *IEEE: Intelligent Systems*, vol. 14, pp. 61-69, 1998.
- [3] A. C. Graesser and N. K. Person, "Question asking during tutoring," *American Educational Research Journal*, vol. 31, pp. 103-137, 1994.
- [4] K. R. Koedinger, "Cognitive tutors," in *Smart machines in education*, K. D. Forbus and P. J. Feltovich, Eds. Cambridge, MA: MIT Press, 2001, pp. 145-167.
- [5] S. Milner, "Thompson Prometric," vol. 2004: Thompson Learning, 2004.