INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Expression Evaluation (EE) Tutor in Java: An Intelligent Tutor

Anjeev Kumar Singh[1], Dr. Pawan Kr. Choudhary[2]

[1]Research Scholar, Dr. B.R.A, Bihar University, Muzaffarpur
anjeevsingh@yahoo.com
[2]Dept. Of Mathematics, Dr. J.M. College, Muzaffarpur
pawankrchoudhary@gmail.com

Abstract: *This paper discuss an Intelligent Teacher called EE-Tutor for teaching Java operator precedence, associativity and expressions evaluation to students in the Faculty of Information Technology Dept. in Bihar University, Muzaffarpur, Bihar. An overview of the EE-Tutor architectural design and user interface will be discussed. According to the success of other similar Intelligent Tutoring System, it is also hypothesized that students will be able to learn operator precedence, associativity, expressions evaluation and gain knowledge more quickly and effectively than students using traditional methods of teaching.*

Keywords: *Artificial Intelligence, Expression, Operator, Precedence, Associativity, Expression Evaluation, Java Programming Language.*

1. INTRODUCTION

An Intelligent Tutoring System (ITS) provides an individual computer-based instruction to students [7-8][12]. These systems emerged from application of Artificial Intelligence techniques to the Computer Aided Instruction (CAI) systems [6]. The difference is that an ITS usually compares the student's work with expert solutions or strategies, models the students probably knowledge of a domain and provides coaching or advice, taking into account what the student's knowledge state and preferred learning style. Depending on Artificial Intelligence (AI) and cognitive science, ITS became very popular and effective domain in Education for many reasons: Better Student Performance, Student learn in Less Time and Student is in the driver seat [5],[11],[15]. For many years, there is a continuous development and evaluation of ITS [18][19] for tutoring and monitoring programming field of Artificial Intelligence in Education. Programming requires a group of problem-solving and diagnostic strategies. The behavior in which a student writes code provides great insight into the way of his thinking. As a result, programming provides attractive area to study learning and cognitive processes [3][12]. Among the objectives of this research is to gather the developments in the ITS, Cognitive Science and AI to make a useful intelligent tutor to help students understand Java Programming Expression Evaluation.

Generally textbooks don't provide the steps required solving a problem, but using visualization [9] and intelligent tutoring systems, student can learn and solve problems comfortably.

2. DESIGN OF EE-TUTOR

2.1. Architecture and Design of EE-Tutor:

We present the architecture and model for the Expression Evaluation Tutor: An Intelligent Tutor, the knowledge base design, expert module design, feedback module design and user interface design.

The EE-Tutor could be part of a courseware on teaching Java programming language to undergraduate and post-graduate students at the university level.

2.2. EE-Tutor knowledge base design:

In this study, we have concentrated on the syntax of expressions, operator precedence, associativity and evaluation of Java Expressions.

Java has well-defined rules for specifying the order in which the operators in an expression are evaluated when the expression has several operators [1],[2],[4].

2.3. Precedence Order:

When two operators share an operand, the operator with the higher precedence goes first. For example, 9 - 3 * 2 is treated as 9 – (3 * 2), where as 9*3-2 is treated as (9*3)-2 since multiplication has a higher precedence than subtraction.

2.4. Associativity :

When two operators with the same precedence, the expression are evaluated according to its associativity.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

For example x = y = z = 7 is treated as x=(y=(z=13)), leaving all three variables with the value 7, since the = operator has right-to-left associativity. On the other hand 50/2/4 is treated as

Precedence	Operator	Type	Associativity
13	( )	Parentheses	Left to Right
12	+ + - -	Unary post-increment Unary post-decrement	Right to Left
11	+ + - - + - | ~	Unary pre-increment Unary pre-decrement Unary plus Unary minus Unary logical negation Unary bitwise complement	Right to Left
10	* / %	Multiplication Division Modulus	Left to Right
9	+ -	Addition Subtraction	Left to Right
8	< <= > >=	Relational less than Relational less than or equal to Relational greater than Relational greter than or equal to	Left to Right
7	= = ! =	Relational is equal to Relational is not equal to	Left to Right
6	&	Bitwise AND	Left to Right
5	^	Bitwise Exclusive OR	Left to Right
4	|	Bitwise Inclusive OR	Left to Right
3	& &	Logical AND	Left to Right
2	| |	Logical OR	Left to Right
1	= + = - =	Assignment Addition Assignment Subtraction Assignment	Right to Left

Fig – 1: Operator precedence and its associativity

(50/2)/4 since the / operator has left-to-right associativity.

We have stored the operators of Java, Its precedence and associativity in the knowledge base.(Fig 1).

For every problem in the knowledge base, we have stored the possible solution, possible errors for specific categories and possible hints for each error. Fig-2 shows a problem example with a solution and some errors.

Problem: Evaluate the expression res in the following Java program segment and find the output of the program

```
class Precendence {
  public static void main ( String [ ] args) {
  int a = 6;
  int b = 5;
  int c = 10;
  int d = 8;

  float res = 0;
  res = a % b * ++c - --d  + 2
 }
}
```

The Solution which is authored by Instructor:

Operator	=	%	*	++	-	--	+
Evaluation Order	7	3	4	2	6	1	5

Operation	Operation Result	Expression
1	7	--d
2	11	++c
3	1	a % b
4	11	a % b ++c
5	9	--d + 2
6	20.0	a % b ++c + --d + 2
7	20.0	res = a % b ++c + --d + 2
Output of the Program		The result is : 20.0

The student first possible error : Associativity problem between ++c and - -d

Operator	=	%	*	++	-	--	+
Evaluation Order	7	4	3	1	5	2	6

The student second possible error : Associativity problem between % and *

Operator	=	%	*	++	-	--	+
Evaluation Order	7	4	3	1	5	2	6

The student third possible error : Precedence problem between - and +

Operator	=	%	*	++	-	--	+
Evaluation Order	7	4	3	1	5	2	6

Fig – 2: A problem example with a solution and a few possible

The problem stored in the knowledge base can by classified into 5 levels of difficulties. Difficulties mean more operators, mixed associativity types and different level of precedence, are involved in the problem. The Expert model determine which problem level should be given to the student.

In order for the student, to solve a problem, he must first determine operator order of evaluation by filling

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

the table as shown above. Second, the student should fill the second table by evaluating each operator at a time and third, put the output of the expression in the last line of the second table as shown in Fig-2.

3. MODULE OF EE-TUTOR

3.1. Expert Module of EE-Tutor:

Expert Module of EE-Tutor is an expert system that is used to determine the student level of understanding from the problem statement, the problem specification and student's answers. When the student reaches a certain score answering at the current level of difficulty; for example 75% or more, the expert system increases the level of difficulties of problems to be given for the student. On the other hand, if the student did not reach a minimum score at certain level say 50%, the expert system branches the student to the tutorial on the subject involved in the problem. When the student finishes the tutorial, the expert system permits the student to go back to the question mode. The score of the student, the level of difficulties and his name are all stored in a database for further analysis in the future when the student comes back and use the EE-Tutor. So, the database reflects the actual level of every student as a result of all previous sessions.

Student Solution:

Operator	=	%	*	++	-	--	+
Evaluation Order	7	3	4	2	6	1	5

EE-Tutor: "There is an associativity problem with ++ and --. The order of evaluation is Right to Left". Do you remember that rule?"
Student: Yes
EE-Tutor makes the correction and proceeds with rest of the operators table

Operator	=	%	*	++	-	--	+
Evaluation Order	7	4	3	2	5	1	6

EE-Tutor: "There is an associativity problem with % and *, The order of evaluation is Left to Right. Do you remember that rule?"
Student: Yes
EE-Tutor makes the correction and proceeds with rest of the operators table.

Operator	=	%	*	++	-	--	+
Evaluation Order	7	3	4	1	5	2	6

EE-Tutor: "There is an associativity problem with – with +. The precedence of + is higher than –" Do you remember that rule?"
Student: Yes
EE-Tutor makes the correction and student continue with the solution of the problem by filling the second table

---

Fig – 3: A dialogue between EE-Tutor and the Student

3.2. Feedback Module of EE-Tutor:

Effectiveness of the system depends heavily upon its feedback timing and style. Timing refers to when the student is given a response to the solution. When the feedback is presented to the student should be governed by what the student have done. Tutors are better than teachers in this respect [14][17] in that they can provide a student with timely feedback better than most teachers.
Using the example in Fig-2, the following dialogue between EE-Tutor and the Student would come up as in Fig.3.

3.3. User Interface of EE-Tutor:

The interface of intelligent tutoring system is a very important factor that we gave it a careful consideration during the design of EE-Tutor. The user interface is based on a presentation format implemented in many popular Integrated Development Environments used by professional programmers [13]. After starting of EE-Tutor application, student gets the working environment of EE-Tutor. An appropriate skill-level problem is selected or the problem that last attempted is presented to the student.
The student solves the problem in the Student Solution window. Once the student fills the operators evaluation order table and he press the Check Solution button, the expert module will determine the appropriate response based on the diagnosis of the student solution. A dialogue will be initiated between the EE-Tutor and the student in case of errors. The expert system module informs the user about each error and ask the user weather he remember the rule of precedence or associativity for that operator. If the student answers indicates that he does not remember that rule, the expert system using pattern matching template analyze the student response and display the operator precedence table. Before the expert module proceeds with the next error, it corrects the previous error for the student. Once all errors are corrected, the user will be given control to fill the table of expression evaluation to find its final answer.
The intelligent feedback of the expert module is sent to the student's output window. The student, at any time, may explicitly request from EE-Tutor to view the solution, exit from the current problem and ask for a new one; furthermore, the student can view his performance based on statistics including problems attempted, problems solved, number of attempts on a problem and problem difficulty. The EE-Tutor user interface is shown in Fig. – 4.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

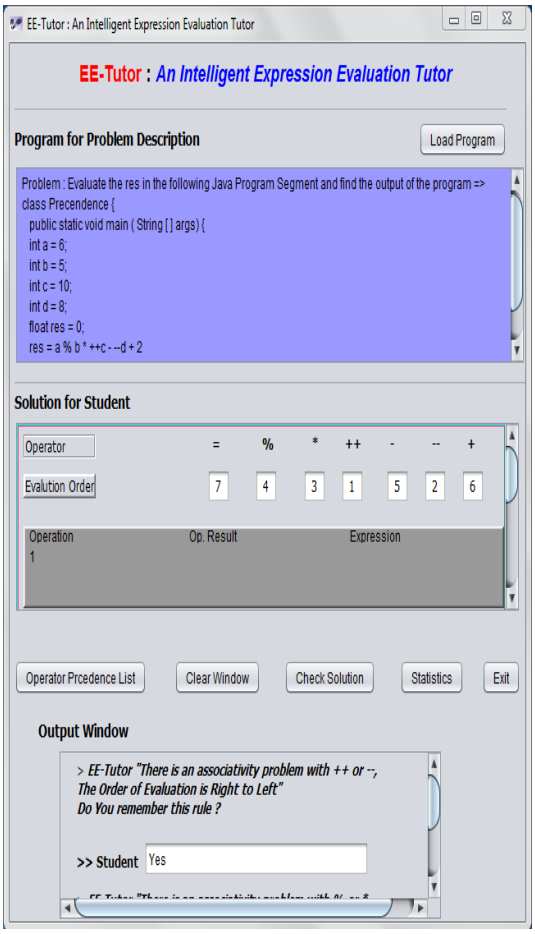

Fig – 4: EE-Tutor User Interface

4. CONCLUSIONS

In this study, we have presented recent developments related to the EE-Tutor Intelligent Tutor for teaching Java Expression Evaluation. EE-Tutor is based on theories, pattern recognition techniques, error detection and correction strategies as in [10],[16],[20]. Even though an evaluation of the EE-Tutor has not been done yet, the implemented examples discussed clearly demonstrate the potential of EE-Tutor. This research is significant since it has the potential to be applied to many programming courses at the university-level.

5. FUTURE WORK

EE-Tutor has not yet been evaluated as to its effectiveness as a tutoring tool. A full evaluation is planned during 2014, to be taken with an introductory Java Programming Class. Additional intelligent tutoring systems for java programming should be invested to enables students overcome their difficulties that are faced in programming language.

References

[1] **Horstmann, 2006. Big Java 2nd Edn, John Willey and Sons.**

[2] **Liang, D.Y., 2008. Introduction to Java Programming- Comprehensive Version. 7th Edn. Prentice Hall.**

[3] **Scott, A.C., J.E. Clayton and E.L. Gibson, 1991. A Practical Guide to Knowledge Acquisition. (Menlo Park, CA: Addison-Wesley).**

[4] **Thomas, W.U., 2006. An Introduction to Object-Oriented Programming with JAVA, 4/e Tata McGraw-Hill Publications.**

[5] **Woolf, B.P., J. Beck, C. Eliot and M. Stern, 2001. Growth and Maturity of Intelligent Tutoring Systems. In: Smart Machine in Education, Forbus, K.D. and P.J. Feltovich (Eds.), Cambridge, MA: MIT Press, pp: 100-144.**

[6] **Abu Naser, S.S. and O. Sulisel, 2000. The Effect of using computer aided instruction on performance of 10th grade biology in Gaza. J.Coll. Educ., 4 (1): 9 - 37.**

[7] **Abu Naser, S.S. 2001. A comparative study between animated intelligent tutoring systems AITS and video-based intelligent tutoring systems VITS. Al-Aqsa Univ. J., 5(1): 72-96.**

[8] **Abu Naser, S.S. 2006. Intelligent Tutoring System (ITS) for teaching database to sophomore students in Gaza and its effect on their performance. ITJ., 5(5) : 916-922.**

[9] **Abu Naser, S.S., 2008. Developing visualization tool for teaching AI searching algotithems. ITJ., 5(5): 916-922.**

[10] **Aleven, V. and K.R. Koedinger, 2002. An effective meta cognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor, Congnitive Science., 26(2): 147-179.**

[11] **Anderson, J.R., A.T. Corbett, K.R. Koedinger and R. Pelletier, 1995. Cognitive Tutor: Lessons learned. JLS., 4: 167-207.**

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

[12] **Brusilovsky, P., E. Schwarz and G.Weber, 1996. ELM-Art: An Intelligent Tutoring System on World Wide Web. In: Intelligent Tutoring Systems, Claude Frasson, Gilled Gauthier and Alan Lesgold (Ed.). Volume 1086 of Lecture Notes in Computer Science, Berlin, Germany Springer Verlag,. pp: 261-269.**

[13] **Conlan, O., C. Hockemeyer, V. Wade, D. Albert and M. Gargan, 2002.An Architecture for integrating adaptive hypermedia service with open learning environments. In: Proceedings of ED-MEDIA 2002, Volumn 1 of World Conference on Educational Multimedia. June 24-29, 2002. Hypermedia and Telecommunications, pp: 344-350.**

[14] **Du Boulay, B., 2000. Can we learn from ITSs? In : Intelligent Tutoring Systems. Proceedings of the 5th International Conferences ITS, Gauthier, G., C. Frasson and K VanLehn (Eds.). Springer-Verlag, Berlin, 1839: 9-17.**

[15] **Graesser, A.C. and N.K. Person, 2001. Teaching tactics and dialog in auto tutor. IJAIE., 12: 12-23.**

[16] **Koedinger, K.R., 2001. Cognitive Tutors. In: Smart Machines in Education, Forbus, K.D. and P.J. Feltovich (Eds.). Cambridge, MA: MIT Press, pp: 145-167.**

[17] **Mark, M. and J.E. Greer, 1995. The VCR tutor: Effective instruction for device operation. J. Learn. Sci., 4(2): 209-246.**

[18] **Shute, V.J. and R. Glaser, 1990. A large-scale evaluation of an intelligent discovery world: Smithtown. Interactive Learn. Environ, 1:51-77.**

[19] **Shute, V.J., 1995. SMART Evaluation: Cognitive Diagnosis, Mastery Learning and Remediation. In: Proceedings of AI-ED 95, Charlottesville. 16-19 August 1995. Greer, J. (Ed.), VA: AACE, pp: 123-130.**

[20] **Sykes, E.R. and F. Franek, 2003. A prototype for an intelligent tutoring system for students learning to program in Java. Proceedings of the IASTED International Conference on Computers and Advanced Technology in Education. June 30-July-2003. Rhodes, Greece, pp: 78-83.**