# A logical and structural thinking development tool (LST) to enhance fundamental problem-solving skills of learners of information technology

**Abstract.** The role of information technology in modern education has increased significantly over the past two decades [14]. The opportunity to develop an interactive software system with the aim of enhancing fundamental problem-solving skills of learners enrolled for the Computer Science, Information Technology and Mathematics programs at tertiary institutions is possible with object-oriented programming techniques and multi-dimensional graphic design. The definition of fundamental problem-solving skills includes cognitive functional skills such as logical thinking, conceptualism with prior knowledge, relationship forming and objective analysis. Experiments done for this research indicate that given the right educational tools, cognitive functional skills of learners can be stimulated, developed and enhanced. This, in turn, may lead to an increase in the graduation rates of learners enrolled for the Computer Science, Information Technology and Mathematics program and ultimately contribute to the reshaping of the educational experience.

**Keywords.** Education, interactive learning, problem solving, object-oriented techniques

## 1. Introduction: Low Graduation Rates

In a report titled *Towards a new higher education landscape: Meeting the equity, quality and social development imperatives of South Africa in the 21st century, South Africa,* the government of South Africa stated that unacceptably large numbers and proportions of students drop out of the tertiary education system. The higher education system's intake of first-time entering undergraduates has averaged about 120 000 for the past few years. At least 30 000 (25%) of these new under-

1 Annelie Jordaan: Faculty of Applied and Computer Sciences, Vaal University of Technology. Prof. Dawid Jordaan: School of Modelling Sciences, North-West University.

graduate students drop out of universities or technikons at the end of their first year of study. The total number of students that drop out of South African universities and technikons (technikons are renamed as Technical Universities) is at least 100 000 students per year, out of an enrolment total of about 600 000 students. Another major aspect of inefficiency in the system is the retention of failing students in the system. A number of institutions report poor success rates by course (averages of 70% and below), and low graduation rates (often 15% or below).

According to the 1996-2001 research report on retention and graduation rates at the University of Kentucky [1], USA, the term *graduation rates* reflects the percentage of full-time learners in the cohort who earned a bachelor's degree (or higher level degree) within a specified number of years. Cohorts refer to classes of first-time, full-time degree-seeking learners. This report indicates that the average 6-year graduation rate (earning a bachelor's or higher degree within 6 years) for first-time, full-time and first-year degree-seeking learners is 57.7%.

In a report by the Association for Institutional Research Forum [2], USA, the results of the 5-year graduation rates of first-time, full-time freshmen at 53 public universities are shown. The graduation rates vary from 25.7% to 77.1%. The average 5-year graduation rate is 54.8%. Significant to this report is the student migration patterns across programs. Of the 112,000 graduates in the database, only 9.3% graduated in the Sciences and Mathematics program compared to Social Sciences with 22.5% and Business, Management & Public Administration with 26%.

At Brock University [3], Canada, a report on the graduation rates for programs during the period 1993-1999 was compiled and published by the Ministry of Training, Colleges and Universities. The 7-year graduation rate for the Computer Science program is 61.8%. In contrast with this is the 7-year graduation rate of 97.6% for the Education program, 80.5% for the Applied Health Sciences and 79.1% for the Social Sciences program.

At the McMaster University [4], Canada, not even one learner in the Computer Science program graduated within the first 5 years after enrollment in 1992. The South African Department of Education [5] stated that South Africa is not excluded from the international trend of a significantly low graduation rate for learners enrolled in the fields of Applied Technology, Engineering and Computer Science. One of the contributing factors to this troublesome situation is the lack of adequate fundamental problem-solving skills of learners enrolled for these programs [7].

## 2. Purpose of this study

The purpose of this paper is to propose an object-oriented, multi-dimensional software tool with the aim of testing, developing and enhancing fundamental problem-solving skills of learners enrolled for the Computer Science, Information Technology and Mathematics program. Experimental results show a significant improvement in the cognitive functional skills such as logical and structural thinking of learners who used this software tool in a pilot study over a period of two months or longer.

## 3. Fundamental problem solving skills

In South Africa the *pass rates* (percentage of learners passing a specific course/module/subject for the first time enrolled) for programming courses at tertiary institutions are problematically low, and the situation is deteriorating. Results from a preliminary study conducted in South Africa at two universities to evaluate the fundamental problem-solving skills of learners enrolled for programming courses, clearly indicate the need for educational tools to enhance their logical, structural and critical thinking [7].

To be logical involves three simple steps: 1) be observant (notice things, data and events), 2) identify a pattern, and 3) reasoning (give a cause for a phenomenon) [9]. Computer Science, Information Technology and Mathematics are three subjects that require an in-depth knowledge of fundamental problem-solving concepts. These concepts include cognitive functional skills such as logical thinking, conceptualism with prior knowledge, relationship forming and objective analysis [8]. Given the right educational tools, fundamental problem-solving skills of learners can be stimulated, developed and enhanced. This, in turn, may lead to a significant increase in the pass rates of learners.

## 4. Object-oriented programming

Objects are the central idea behind object-oriented programming, and the basic idea behind an object is that of simulation. Most programs are written with very little reference to real-world objects. In object-oriented methodology, a program should be written to simulate the states and activities of real world objects. When modeling an object, the methods associated with that object must also be taken in consideration. A method is an operation that can modify an objects' behavior. In other words, it is something that will change an object by manipulating its variables [11]. One of the principal advantages of object-oriented programming techniques over conventional programming techniques is that it enables programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. In addition, programmers can create relationships between one object and another. This makes object oriented programs easier to modify [12].

## 5. Logical and structural thinking development tool (LST)

Modern technology provides us with the tools we need to create a more dynamic, interactive environment for learners [10]. Object-oriented programming techniques and multidimensional graphic design have unlocked the opportunity of developing an interactive software system (LST – Logical and Structural Thinking development tool) to enhance fundamental problem-solving skills of learners enrolled for programmes in Computer Science, Information Technology and Mathematics. Techniques and object-oriented concepts used in developing the LST development tool include two- and three-dimensional graphics, simulation, dynamic linking (making code re-usable to avoid re-implementation), encapsulation (information hiding), artificial intelligence (complex decision making and intelligent reasoning) and file encryption and decryption.

### *5.1. LST design structure*

The LST development tool consists of two separate modules: the question Developer and the question Viewer. The facilitator uses the Developer to design and develop any number of logical and structural thinking questions within specified boundaries. These questions are saved to a file structure. The learner does not have any access to the Developer module. The Viewer is used to display the questions and potential answers to the learner. Each question has one or more hints to guide the learner towards the correct answer. The learner has the choice of completing an electronic test without the aid of hints, or simply working through one or more of the question categories using the available hints. In either case the score of the learner is written to a database to enable the facilitator to keep track of the learner's progress. LST can function on both a standalone personal computer and a local area network. Assessments can be completed electronically or in printed format.

The layout and design structure of the LST development tool consist of the following components.

1. 3D Rendering Engine [NON-VISUAL]: Manages the scaling, rotating, sorting, aligning and drawing of all 3D objects.
2. 3D Object Designer [VISUAL]: Provides a visual link with the non-visual 3D rendering engine. Contains all the necessary tools to change colors, positioning 3D cubes, show wire frames and add labels in an easy to use format. Provides the user with total 3D development of a

square grid, circle, cube, pyramid and diamond.

3. 3D Rotating Engine [NON-VISUAL]: Provides the user with the tools to rotate the created 3D model in all three dimensions (X, Y, Z-axis).
4. 2D Drawing Editor [VISUAL]: Provides a linkage between the local file system and the editor itself. Provides a linkage to the 3D Object Designer for the retrieving of 3D images. Provides basic drawing tools. Has built-in image scaling features.
5. I/O System [NON-VISUAL]: This non-visual class provides an interface with the local file system to store all question-related information in encrypted format.
6. TANGRAMS Engine [NON-VISUAL]: This engine enforces the rules defined by the TANGRAMS concept.
7. TANGRAMS Display [VISUAL]: Does all the drawing and editing options of the TANGRAMS Engine.
8. Question, Answer and Hint Interface [VISUAL]: Question Interface, Answer Interface, and Hint Interface. All three interfaces provide a format and layout for the respective areas, thus, how the questions, answers and hints will be displayed.
9. Summary Display [VISUAL]: Provides easy access to the questions and displays all the information of the question list.
10. Test Compiler Interface [VISUAL]: Provides the user with a compiler to assemble a list of questions. Used to test the progress of the student.
11. Development and Design Interface [VISUAL]: Provides a controlled environment to display all information.

## *5.2. LST question categories*

The questions built into the LST development tool can be divided into four main categories. Each main category can contain any number of subcategories. The degree of difficulty for each sub-category can be set to easy, medium or difficult.

### ***5.2.1 Logic and mathematics***

This main category allows the facilitator to develop questions in plain text format. Images and 3D objects may be included in the questions, and the answer format may consist of five short-answers, five images or five 3D objects. The hint and answer formats are the same. Although the system allows for any number of subcategories to be defined, the current question file system consists of the following three sub-categories.

- **Mathematics – text**

*Example.* A factory manufactures kitchenware. 6 Electric knives are manufactured for every 3 electric kettles. In a month the factory produces a total of 684 electric knives and kettles. How many of these are kettles?

a) 114

b) 456

c) 570

d) 228

e) None of the above

- **Mathematics – Sequence:**

*Example.* Determine the next number in the series and select the correct answer:

4, 12, 16, 26, 28, 40...

a) 40

b) 38

c) 54

d) 28

e) 0

- **Logic – Sequence:**

*Example.* What comes in the place of the question mark in the following sequence?

ooooo+oooo-ooo++oo--o**?**

a) o+++

b) o---

c) +++

d) ---

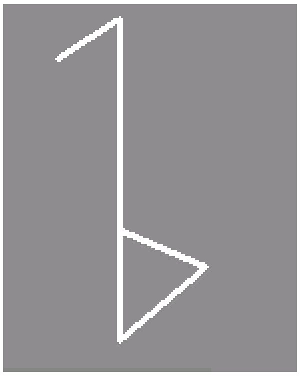
e) o

### *5.2.2 2D Area and logic*

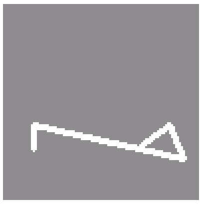
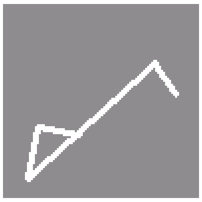
This category's aim is to test, develop and enhance the two-dimensional thinking of the learner. Most of the questions will be compiled using plain text and a bitmap image, although an image and text or 3D objects with short text are also allowed. The answer and hint formats are the same for all categories. The current question file system consists of the following three subcategories.

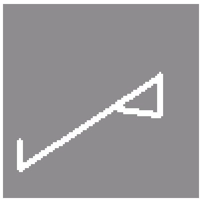
- **Spatial**

*Example.* The question consists of a number of figures - the given figure and five possible answers. If one of the five possible answers is kept flat and turned to the left or right side only, it will be exactly similar to the given figure. Find the answer.



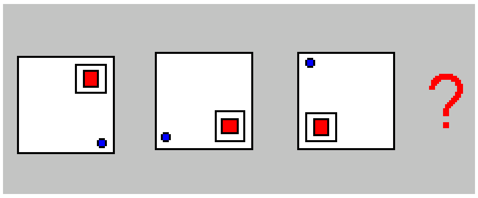
a)  b) 

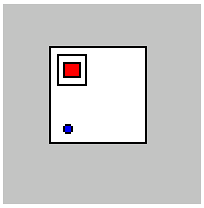
c)  d) 

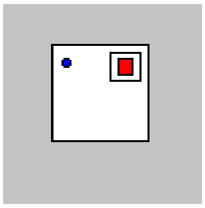
e) 

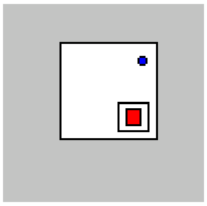
- **Sequence**

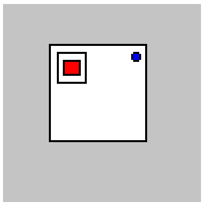
*Example.* What would the next picture in this series be?

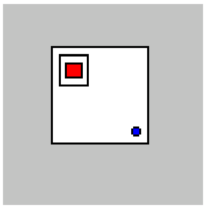



a) 

b) 

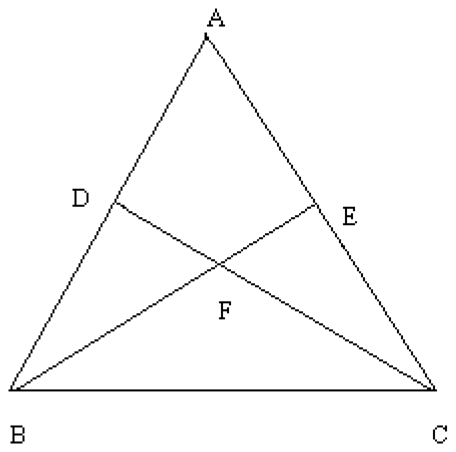
c) 

d) 

e) 

- **General**

*Example.* How many triangles are formed in this figure?




a) 4
b) 5
c) 6
d) 7
e) 8

### *5.2.3 3D Volume and logic*

The aim of the 3D Volume and Logic category is to test, develop and enhance the three-dimensional thinking of the learner. A various number of questions, answer and hint formats are possible. It can be summarized as follows (any permutation of these formats may be combined into a single question.).

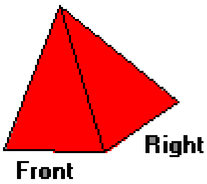
| **Question format** | **Answer format:** |
|---|---|
| Full text | Five short answers |
| Image and text | Five images |
| 3D object and short text | Five 3D objects |

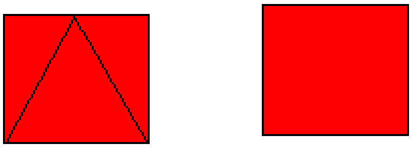
| **Hint format** | **Degree of difficulty:** |
|---|---|
| Full text | Easy |
| Image | Medium |
| 3D object | Difficult |
| None | |

The current question file system consists of the following two sub-categories.

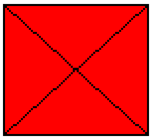
- **Views**

*Example.* Given a pyramid as seen from the front right corner. Find the top view.

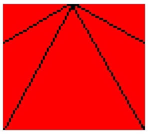


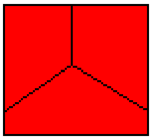
a) b)



c)



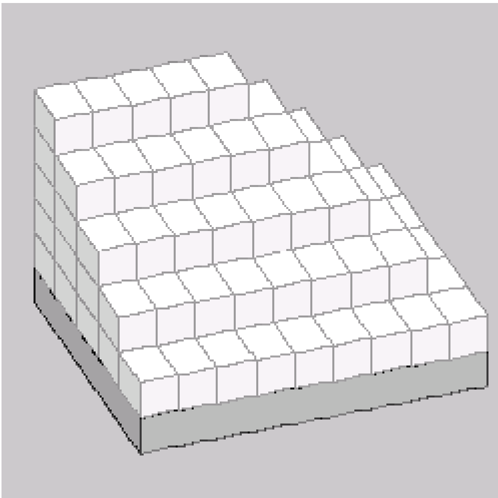
d)



e)

- **Spatial**

*Example.* How many white blocks are in the pile below?



a) 115

b) 85

c) 45

d) 135

e) None of the above

### 5.3.4 Tangrams

A tangram is an ancient Chinese moving piece puzzle, consisting of 7 geometric shapes. The tangram is a square that is cut in such a way that it gives you two big triangles, one medium triangle, two small triangles, one square, and one parallelogram. They can be arranged to make interesting shapes [13]. The LST development tool uses the basic idea behind tangrams, but any number and combination of the given geometric shapes may be used to build a puzzle. Also, it is not compulsory for a puzzle to contain each of the 7 shapes.

The outline of the puzzle is displayed to the learner, with each of the geometric shapes required to fill the puzzle neatly arranged next to the outline. The learner has to drag, and rotate if necessary, each shape into its correct position. Where more than one solution is possible for the same puzzle, LST is intelligent enough to recognize and acknowledge these solutions.

Hints are available to the learner. Each time the learner requests a hint, one piece of the puzzle is randomly selected and displayed. An intelligent randomizer is used to avoid the display of the same puzzle piece twice.

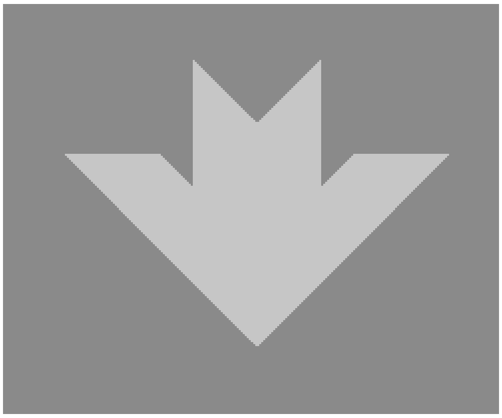


**Figure 2: Tangram outline – spaceship**

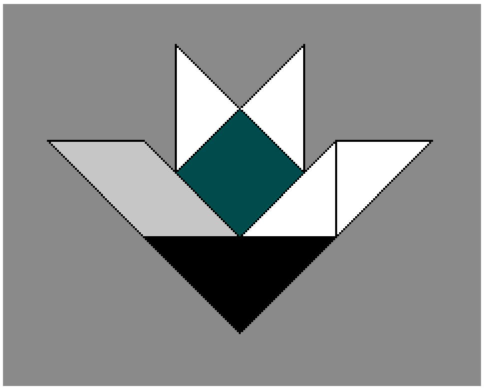


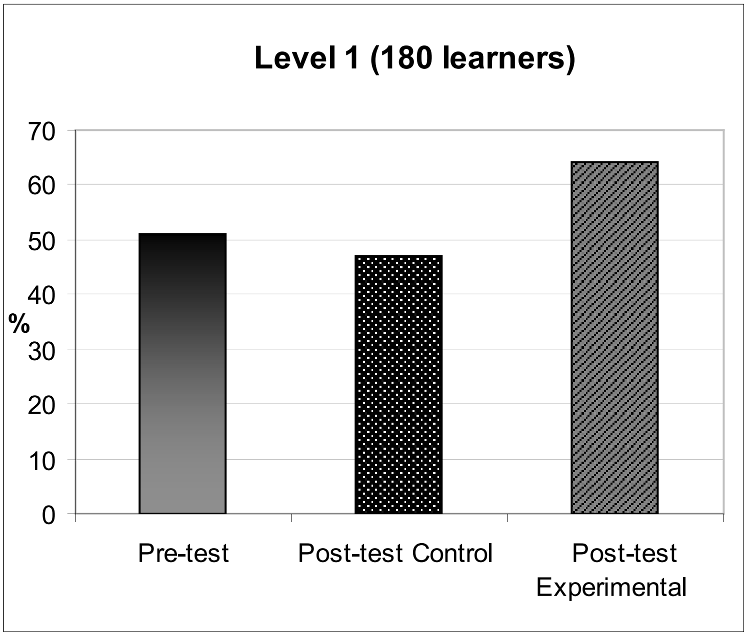
**Figure 3: Solved tangram - spaceship**

## 6. Experimental results

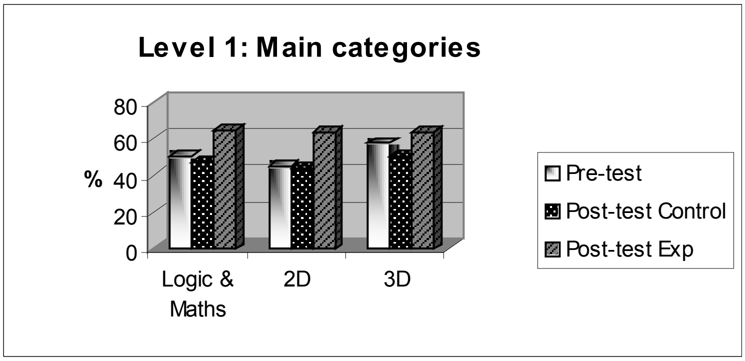
The objective of the pilot study was to determine whether the LST development tool could successfully be used to develop and enhance fundamental problem-solving skills of learners. All first, second and third level Information Technology learners at the Vaal University of Technology were involved in pre-post tests from August – October 2003. These learners completed a printed LST test during the last week of August. Afterwards the learners of each level were randomly subdivided into a control group and an experimental group.

Learners in the three control groups completed another printed LST test during the last week of October. The two tests contained similar questions, four from each of the subcategories, a total of 32 questions. Learners in the three experimental groups also completed the second printed test, but in addition they used the online LST Viewer to do exercises in all question categories on a weekly basis, 90 minutes per week for eight weeks. The third level experimental group used LST for ten weeks. The Viewer was used in the period between the writing of the two tests. The results are shown in the following graphs.

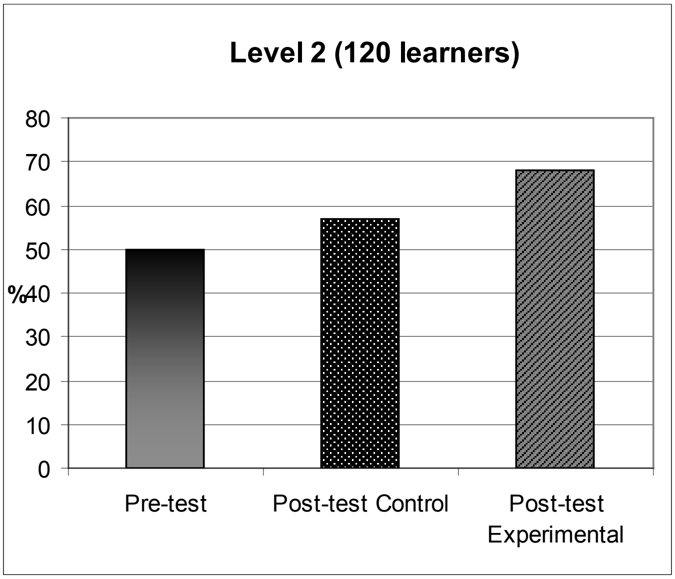
**Graph 1: Pre-post test for level 1 learners**



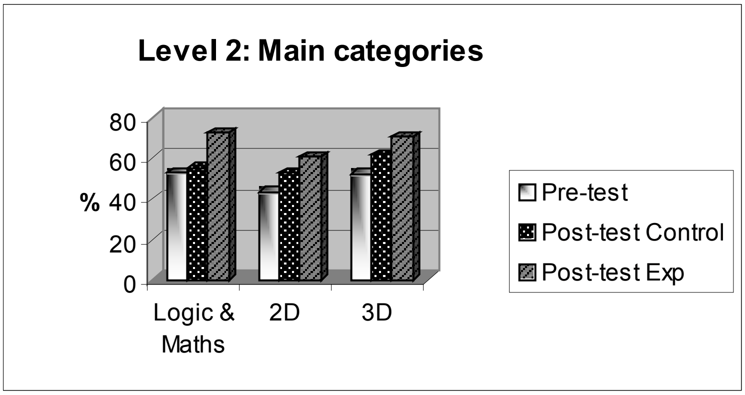

| | Logic & Maths | 2D | 3D |
|---|---|---|---|
| Pre-test | 50.51 | 44.91 | 57.35 |
| Post-test Control | 46.83 | 43.43 | 50.53 |
| Post-test Exp | 64.38 | 62.92 | 63.44 |
| **STDEV** | **9.25** | **10.85** | **6.46** |



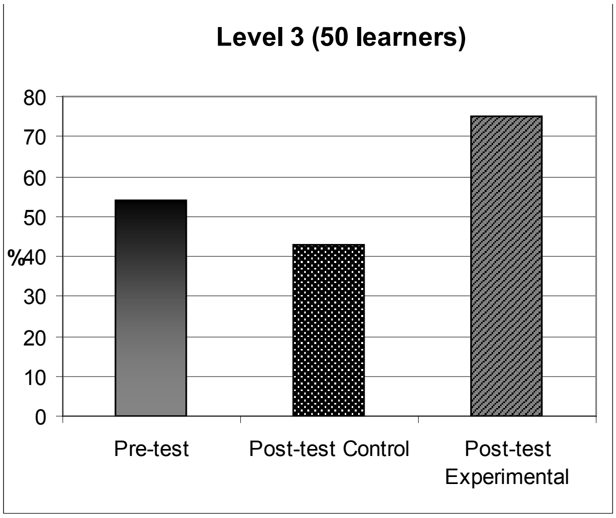
**Graph 2: Pre-post test for level 2 learners**



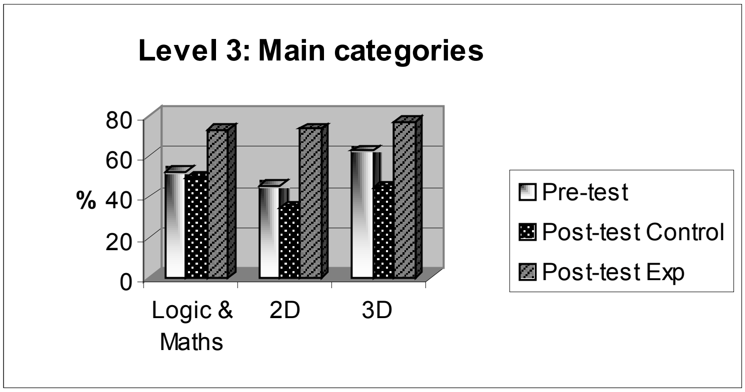

| | Logic & Maths | 2D | 3D |
|---|---|---|---|
| Pre-test | 52.66 | 43.72 | 51.99 |
| Post-test Control | 55.3 | 52.78 | 61.74 |
| Post-test Exp | 72.32 | 60.71 | 70.54 |
| **STDEV** | **10.67** | **8.50** | **9.28** |



**Graph 3: Pre-post test for level 3 learners**




| | Logic & Maths | 2D | 3D |
|---|---|---|---|
| Pre-test | 52.56 | 45.73 | 62.5 |
| Post-test Control | 49.48 | 34.9 | 44.54 |
| Post-test Exp | 72.83 | 73.55 | 77.18 |
| STDEV | **12.69** | **19.94** | **16.35** |



## 7. Conclusion

The graphs clearly indicate that the results of the learners in the experimental groups improved significantly from test 1 to test 2 on all three levels. The experimental group on the third level showed the highest increase: from 54% for test 1 to 75% for test 2. It is also this group of learners who used LST for two weeks longer than the other two experimental groups. The results of the control groups show a decrease on level 1 and 3, and an increase on level 2. Variables to be considered might be a slightly more difficult second test, an above-average intelligent second level group and the random assignment of groups.

We conclude that, with the frequent use of the LST Viewer, fundamental problem-solving skills of learners may be stimulated, developed and enhanced. This, in turn, may lead to an increase in pass rates of learners enrolled for the Computer Science and Information Technology program at academic institutions.

## References

[1] University of Kentucky, *Retention and graduation rates entering classes of 1986 to 2001*, USA, 2001, 1-2. [On-line]. Available: http://www.uky.edu/IR/dsi/retention/retgrad02.pdf

[2] R. Kroc, R. Howard, D. Woodard & P Hull, *Graduation rates: Do students' academic program choices make a difference?,* Association for Institutional Research Forum, Orlando, USA, 1997, 3-12. [On-line]. Available: http://aer.arizona.edu/AER/Enrollment/Policy_Analyses/GradRates.PDF

[3] Brock University, 1992 Graduation rates, *Performance indicators - Office of Institutional Planning and Analysis,* Ontario, Canada, 2001. [On-line]. Available: http://www.brocku.ca/instanalysis/performanceindicators/2001/graduationRates.html

[4] McMaster University, *Graduation, employment and OSAP loan default rates*, Canada, 2001. [On-line]. Available: http://www.mcmaster.ca/ua/opr/osap/programtable01.html

[5] Department of Education (DoE), *Towards a new higher education landscape: Meeting the equity, quality and social development imperatives of South Africa in the 21st century,* South Africa, 2000. [On-line]. Available: http://education.pwv.gov.za/DoE_Sites/Higher_Education/CHE/CHE_Report30June2000.htm

[6] Vaal University of Technology, *Pass rates 1997-2002*, Vanderbijlpark, South Africa, 2003.

[7] D.B. Jordaan, Research findings: *Evaluating fundamental problem-solving skills of learners enrolled for a programming course*. Vanderbijlpark, South Africa, 2000. (Unpublished).

[8] P. Engelbrecht, S. Kriegler, & M. Booysen, Perspectives on learning difficulties, international concerns and South African realities (Pretoria: J. L. van Schaik, 1996).

[9] E.W. Teck, *The skills of logical thinking,* Singapore Institute of Management, Singapore, 2001. [On-line].
Available: http://www1.sim.edu.sg/sim/pub/mag/sim_pub_mag_list.cfm?id=862&sterms=Logical|thinking

[10] C. Howell, *Teaching history online (or creating a dynamic past from a technological present)*, Proc. International Online Conference on Teaching Online in Higher Education, Indiana, USA, 1999 [On-line].
Available: http://as1.ipfw.edu/99tohe/presentations/howell1.htm

[11] A. O' Neill, *Object-oriented programming: A guide for the beginner – from Modula-2 to Java*, 1999. [Online].
Available: http://www.quiver.freeserve.co.uk/OOP3.htm

[12] Wibopedia, *Object-oriented programming, the #1 online encyclopedia dedicated to computer technology*, 2002. [On-line].
Available: http://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html

[13] L. Dano, *Exploring Tangrams – an interactive web sampler,* James S. Daley Middle School, MA, USA, 2002. [On-line].
Available: http://dal.lowell.smartedu.net/projects/Tangrams/Tangrams.html

[14] P Jen-Hwa Hu, T.H.K. Clark, W.W. Ma, *Examining technology acceptance by school teachers: a longitudinal study*, Information & Management, 41 (2003) 227-241.