EXAMINING THE RELATIONSHIP BETWEEN ABSTRACT THINKING ABILITY AND ACADEMIC PERFORMANCE IN OBJECT ORIENTED PROGRAMMING: A PSYCHOMETRIC AND THEORETICAL ANALYSIS OF STUDENT CAPACITY

M. KOMPI & L. GROBBELAAR
CENTRAL UNIVERSITY TECHNOLOGY, FREE STATE

Abstract

Abstract reasoning involves flexible thinking as well as solving problems creatively and logically. Evidence suggest that especially first year students that undertake Information Technology studies at institutions of higher education, find it challenging to successfully comprehend concepts regarding object oriented programming (OOP). This study is based on the hypothetical premise that student performance regarding OOP can be linked to the abstract thinking ability of the particular student because of the highly abstract nature of OOP concepts. A comparison between psychometric test results gauging abstract thinking abilities of students and student performance in OOP are also presented and argued.

Keywords: Abstract thinking, abstract reasoning, object oriented programming, psychometric testing, student performance

1. INTRODUCTION

Extraordinary works of art, mathematics and physical science are but a few examples of what is possible through the ability of the human being to think and relate objects in an abstract fashion. Abstract thinking ability has been credited as being a key role player and talent amongst many cognitive factors that contribute towards creativity, innovation and problem solving abilities. At this stage many arguments exist with regard to basic education and training systems and if it is doing enough to produce fresh thinking leaders in the fields of mathematics, science and engineering. This is especially true in the South African context.

Object Oriented Programming (OOP) has, in recent years, become a very influential computer programming paradigm. Compared to the structured or procedural programming paradigm, OOP aimed to provide a more natural and intuitive way to conceptualize programming concepts by relating or describing these concepts in terms of objects with associated attributes and instances (Zhang, 2010). OOP is very abstract in nature and requires abstract cognition skills just like, for example, mathematics, when compared to the structured programming paradigm.

Several programming languages that implement the OOP paradigm, for example Java, facilitates class creation via abstract data types (ADTs), which hide their implementation from clients or other classes. ADTs addresses the problem of rewriting code if the data implementation changes, by providing the implementation-independent interfaces to their clients.

As mentioned earlier in this text, both mathematics and OOP have abstract foundations. The ability to think in an abstract manner may therefore impact the academic performance in these subjects positively or negatively.

The scrutiny of math results of students applying to study in the field of information technology (IT) at South African tertiary institutions is no coincidence. It is a gauging mechanism employed by these institutions in an attempt to assess if students who apply will be able to perform well in the relevant field of study. As OOP is growing in popularity amongst programming professionals, tertiary institutions facilitating education and training in IT have adopted curricula that include intensive OOP training, albeit in a time constricted fashion.

Abstract thinking ability can be assessed from a psychological perspective. Various assessments and instruments have been developed and tested throughout recent human history with the aim to investigate and quantify human cognitive capacity in its different forms. Psychometrics is one of the fields in which these assessments and instruments are utilized on a regular basis.

This paper aims to present and disseminate findings regarding research done by the authors that relates to the abstract thinking abilities of first year students at a tertiary academic institution in South Africa studying in the field of IT and the academic performance of these students in OOP. A psychometric lens was employed to observe the abstract reasoning skills of the named students.

2. LITERATURE REVIEW

Mathematics is a subject that is universally applied in many disciplines such as astrology, engineering, computer programming, theoretical physics and the like. From the school benches to the university lecture halls, mathematics plays an important role in daily life.

According to researchers such as Sinaceur (2014), the “mathematical thinking ability” empowers problem solvers who possess such skills with greater problem solving abilities, depending on their particularities, over individuals whom lack the relevant skill or ability.

In fact, evidence from various sources of literature indicate that the ability to successfully comprehend and apply mathematical constituents such as, for example, the symbolic expression of relative values, numerical reasoning, cognitive computing, abstraction, generalization and formalization does and can contribute towards extending the limited powers of the human mind (Armoni, 2013).

Early work done by Varma and Schwartz (2011) described the process of learning mathematics as abstraction over perceptual-motor representation into richer and more abstract concepts. Furthermore Varma and Schwartz (2011) continued by stating that integers inherit natural or positive numbers, zero and negative numbers, which are all, in fact, abstract numbers. This notion was strongly supported by Sinaceur (2014).

Early studies by Avraamidou and Monaghan (2009) suggested that the mathematical symbols which indicate addition, subtraction, multiplication (+, -, x) etc. can be grouped with other mathematical symbols as so-called mathematical abstract concepts. Sinaceur (2014) also asserted that making use of the symbols such as "+" for addition, for example, whilst selecting and assigning properties and operations for these symbols to form different mathematical concepts lead to greater mathematical abstraction.

Object-Oriented Programming (OOP) is, in essence, strongly rooted in mathematics and is therefore also relatively highly abstract in nature. OOP is a departure from the structured or procedural programming design paradigm in many ways, of which the manner in which data values are stored is most prominent. The OOP paradigm makes use of the concept of "objects" consisting of different attributes that will ultimately contain the relevant data. Abstraction, Inheritance, Polymorphism and Encapsulation forms the four fundamental principles of the OOP paradigm (Amstrong, 2011; Cankaya, Yuksel & Arif, 2015). The work done by Armoni (2013) shows that when considering abstraction in the context of OOP, it essentially means to work with abstract classes, declaring and instantiating objects of those classes and extending the incorporation of class attributes through polymorphism and inheritance. All this whilst enabling interaction between objects through encapsulation.

The problem that many students who are exposed to OOP for the first time encounter is the one of abstraction or the ability to think in an abstract manner regarding concepts like objects, attributes, polymorphism and so forth. Also, as Farell (2010) indicated, many students try to relate objects to real world objects as well as how they operate in the real world in an attempt to make sense of the abstract nature of the object. This also contributes to students experiencing difficulty when it comes to trying to comprehend OOP and the relevant OOP concepts.

There are several barriers with regard to learning programming, especially in OOP, which includes a greater nature or complexity. This often involves learning and adapting new ways to conceptualize things as well as a shift in the way one currently thinks and studies. To accomplish this, students require a high level of abstraction ability (Rogerson & Scott, 2010).

Research has indicated that first year students struggle to understand the teaching approach as utilized by lecturers at the tertiary institution level (Mudhovozi, 2012). Similarly, first year IT students who encounter OOP concepts as part of their studies are faced with significant challenges. The study by Alexandron, Armoni, Gordon and Harel (2014) asserted that the fact that OOP requires higher levels of abstraction makes it very complicated for novices. As the evidence suggest, students that possess lower levels of abstraction find it difficult to understand how to represent programming concepts in terms of objects (Armoni, 2013). Moreover, early work done by Al-Linjawi and Al-Nuaim (2010) revealed, amongst other problems, that students do not fully comprehend the OOP development environments, which is also abstract in a sense and that the limited time to cover many OOP concepts in class also does not support their performance positively.

Areas such as psychology, psychiatry, psychometrics and the like have studied comprehension techniques for a duration of time now. Brain techniques such as transcranial direct current stimulation (tDCS) as well as transcranial magnetic stimulation (TMS) are some examples of the tools developed and employed to enhance abstract thinking ability and comprehension. Early studies done by Hamilton, Messing and Chatterjee (2011), showed that these tools increased the cognitive performance of participants on a variety of neurological and psychiatric levels (Hamilton, et al., 2011).

Another study by Coolidge and Overmann (2012), found that numerosity, described as the ability to think and reason with numbers, may have greatly assisted with the development of the human cognitive ability for basic abstractive thinking. Furthermore, the study indicated higher levels of abstract understanding and representation. Similarly, the latest study by researchers Rute-Perez, Santiago-Ramajo, Hurtado, Rodriguez-Fortiz and Caracue (2014), discovered that abstract thinking ability, if cultivated can lead to greater performance in subjects that are abstract in nature like mathematics and so forth (Rute-Perez, et al., 2014).

Subsequent to engaging more literature thoroughly, it was uncovered that a great breakthrough was discovered with regard to how the brain adapts and change during learning, acquiring as well as storing of new information by creating new links between neurons. This was termed motor training-induced neuroplasticity (Bezzola, Merillat & Jancke, 2012).

In fact, neuroplasticity refers to human brain's natural ability to change its structure and form new connections throughout lifespan as a result of one's life experiences (Perwej & Parwej, 2012). Furthermore, amongst different parts of brain are neurons, which represent certain concepts in the brain called grandmother cell. These neurons work together towards accomplishing certain tasks such as completing patterns, functions, tasks and goals. The human brain is a complicated communication system, however, according to Perwej and Parwej (2012) it is capable of solving complex problems within a short period of time, after which the solution in the form of concentrated chemical substances is stored in the synapses of the neurons. The human brain constantly learns how to learn (Acharya, Shukla, Mahajan, 2012). In addition, the study by Klingberg (2010) explains that training the brain's neurons improves performance in wide range of functions and the improved performance is associated with neuronal changes from intracellular levels to the functional organization of the cortex. The research also indicated that abstract thinking abilities in the form of working memory could be stimulated and improved upon.

From the perspective of the psychometric sciences, investigated literature shows that there are indeed several testing instruments used by professionals in this field to assess amongst other things, and not limited to, abstract thinking abilities of people.

Work by Claassen, du Toit, Hoffman and Vosloo (1987) resulted in the development of the manual for an instrument named the General Scholar Aptitude Test (GSAT). This test partly evolved from a comprehensive revision of the New South African Group Test (NSAGT), supported by Human Sciences Research Council (HSRC).

This instrument consists of both verbal and non-verbal (abstract thinking) subtests. Verbal reasoning are examples of subtests assessing non-abstract thinking abilities, while, on the other hand, number series and pattern completion are examples of non-verbal subtests assessing abstract thinking ability (Claassen, et al., 1987).

Our hypothesis was formulated from the literature that was reviewed; it is our hypothesis that abstract thinking ability and student performance in OOP at a tertiary institution, especially on first year level, can be linked.

3. METHODOLOGY

Student performance data regarding assessment results that covered both structured programming and OOP respectively (over a period of 5 years) was collected, scrutinized and quantified. It was found that the average mark for assessments that covered structured programming principles amounted to 57% over the named period. In contrast, the average mark for assessments that covered OOP principles amounted to only 32%.

It is worthwhile to mention here that these averages are specific to the institution where this study was piloted, but clearly indicates a large difference in the average mark of the investigated assessments.

The need to engage a psychologist with skills in the field of psychometrics became a necessity as our aim as to extrapolate the link between abstract thinking ability and the performance of fist year students in OOP in a scientific, known and recognized manner. A registered psychologist was approached whom in return, agreed to assist us and act as custodian and the analyst of the gathered data. Intensive meetings with the psychologist followed in which the various aspects of abstract thinking ability and methods to assess the level of such thinking in individuals were discussed at length.

The researcher was introduced to the psychometric measuring instrument known as the General Scholastic Aptitude Test (GSAT). This instrument is used by qualified psychometric professionals to assess academic abilities in people, amongst other things (Claassen, et al., 1987). According to Claassen, et al. (1987), the GSAT has the versatility to be designed in such a manner that it enables a relevant skilled professional to determine the abstract reasoning abilities, as well as problem solving skills of a particular test.

During the discussions with the engaged professional, the researcher was further briefed on how the psychometric testing will be administered. Considerations regarding different subtests related to testing of none-abstract thinking abilities, important towards overall the results of the test were also discussed. Most importantly, subtests related to the assessment of abstract thinking abilities were deliberated, as was the effectiveness of the test.

All of these important discussions were treated as strictly confidential. Strict measures were carefully employed under the supervision and guidance of the employed professional during the process of approaching and selecting voluntary participants related to this study (and the larger study). This was done to ensure that participants were indeed first year students, undertaking OOP as a subject for the first time and also that all human rights issues were adhered to.

A total of sixty four participants availed them for the assessment process and signed the relevant consent forms. The figure below graphically depicts the racial distribution amongst the participants. A total of sixty one African participants, two Caucasian participants and one mixed race participant formed the constituents of the group.

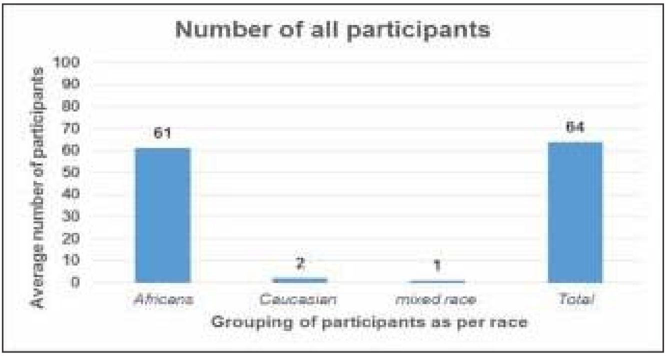

Figure 1: The racial distribution numbers of the participant group

34% of candidates were twenty years of age and 19% of the candidates were nineteen years of age. Whereas 16% of candidates were twenty one years of age and 9% were eighteen years of age. Gender statistics were also recorded, but is not relevant to our research at this stage.

Table 1 provides an overview of both the age range and total number of participants

Table 1: Overview between age range and total number of participants

Age range	Total Number in percentage (%)
21	16%
20	34%
19	19%
18	9%

The assessment was administered in a venue on the campus of a tertiary institution in South Africa and according to the procedures stipulated for such test. This was done, again, under the supervision and guidance of the relevant psychologist employed. The test consisted of sections of subtest. Each subtest was time constrained, which students had to adhere to. Fortunately the psychometric professional also acted as the time-keeper and ensured that the time restrictions were adhered to. The overall duration of the psychometric test was approximately two hours, taking into consideration the stoppages of time in-between each of the subtests in question.

At the conclusion of the assessment, all GSAT instruments used as well as the consent forms were collected from the participants by the psychometric professional with the assistance of the researcher. The assessments were graded and analysed by the psychometric to ensure validity and reliability of the outcomes and the resultant data.

Finally, the participants were subjected to a second assessment compiled by the researcher. The questions of this assessment were compiled to indirectly cover basic OOP and structured / procedural programming concepts. The aim of administering this assessment was to extract the participants' performance and comparing the results of each participant's score with the score obtained in the psychometric assessment.

4. FINDINGS AND DISCUSSION

An analysis of the two assessments as discussed in the previous section as well as the interpretation and comparison of the data collected lead to the findings presented in the section that follow.

4.1 Analysis of the two administered assessments

4.1.1 Student performance in OOP vs Structured Programming

An analysis of the second assessment administered, indicated that only 27% of the students that were evaluated could score above 40% on average with regard to questions covering OOP and OOP related content. In contrast, 68% of the students scored above 40% with regard to questions linked to structured programming. Figure 2 presents the data.

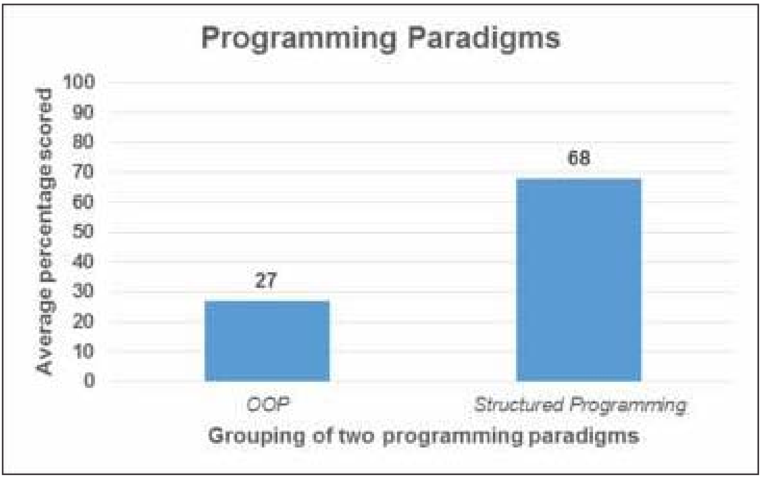

Figure 2: Average percentages scored categorised according to paradigms

4.1.2 Reasoning Ability

The reasoning ability, according to the objectives of the psychometric test (the first test), was divided into two categories: Verbal and non-verbal reasoning. Relevant to this study, the non-verbal reasoning abilities represents the abstract thinking or abstract reasoning abilities of the focus group. Figure 3 graphically represents the reasoning ability distribution of the focus group.

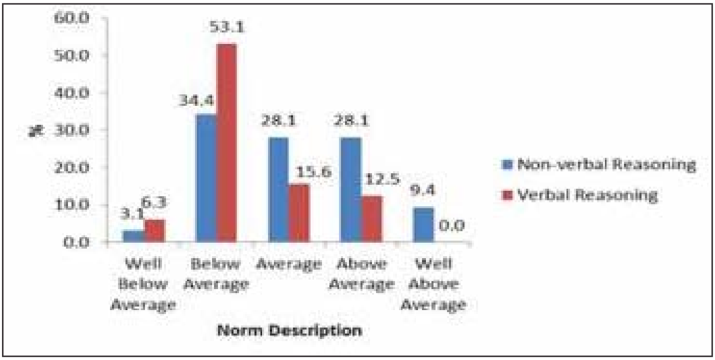

Figure 3: Reasoning Distribution

On average, approximately 66% of the focus group exhibited average to below average abstract thinking abilities with the remaining 34% exhibiting above- to well above average abstract thinking abilities.

4.1.3 Correlation of findings

Table 2 provides an overview of the comparison of the data collected from the two assessments with regard to abstract thinking abilities and student academic performance in OOP.

The data in Table 1 relates the following: Of the sample population 69% received a mark of between 0% and 40% for the OOP related questions in the second assessment. The figure of 69% that scored less or equal to 40% for the named section in the second assessment, appears to support our hypothesis that abstract thinking ability can be related to academic performance in a subject covering OOP concepts and methodologies. This is corroborated by considering that 66% of the sample population scored “average to below average” in the psychometric tests.

Table 2: Comparison between psychometric performance and OOP assessment performance

	Sample Population in % (Rounded)	Performance Rating / Result
Assessment 1 Results: Abstract Thinking Abilities	66	Average to below average
	34	Above to well above average
Total:	100	
Assessment 2 Results: Academic Performance Related to OOP Specifically	27	0% >= 20%
	18	20% >= 30%
	24	30% >= 40%
	21	40% >= 50%
	5	50% >= 60%
	3	60% >= 70%
	0	70% >= 80%
	2	80% >= 90%
	0	90% >= 100%
Total:	100	

5. CONCLUSION AND FURTHER RESEARCH

In this paper, we compared OOP to mathematics as supported by the literature earlier in the text, in the sense that OOP is also a field where the ability to comprehend ideas on an abstract level is relevant and necessary. Moreover, our investigation revealed that, amongst challenges that students face with regards to OOP as a part of their IT studies, they may not fully comprehend the OOP development environments. This is because the environment itself may also be abstract in a sense. Secondly, that the limited time to cover many OOP concepts in allotted classes also does not support student performance in OOP positively.

Our hypothesis that have been put forward for discussion in this paper contends that we can link a student's abstract thinking ability to his or her anticipated performance in subjects that cover or include OOP concepts and methodologies.

The evidence, at least at this stage of the investigation as part of a larger study, corroborates the above mentioned notion. In our view, the findings presented in this paper can be researched further and employed at a more mature stage by tertiary or even secondary educational institutions to identify the abstract thinking abilities of an individual and apply relevant remedial action where necessary. The recent discovery of the brain's Nero-plasticity means that the brain can very well be trained to improve upon current abstract thinking abilities and skills. This will not only aid the individual in question in the school or university benches, but may very well have a larger socio-economic impact in the long run.

In conclusion: The evidence presented here suggests that we can link a first year student's academic performance with regard to Object Oriented Programming to his or her ability to think in an abstract manner. Further research is definitely merited. Age and gender considerations may form part of future models relevant to this study. In the case of the current direction of the study, we aim to investigate, conceptualize and design a comprehensive abstract thinking remedial instrument to address the development of abstract thinking ability(s) in people.

6. REFERENCES

Acharya, S., Shukla, S., Mahajan, S. & Diwan, S., 2012. Localizationism to Neuroplasticity---The Evolution of Metaphysical Neuroscience. JAPI, Volume 60, pp. 38-46.

Alexandron, G., Armoni, M., Gordon, M. & Harel, D., 2014. Scenario-Based Programming: Reducing the Cognitive Load, Fostering Abstract Thinking. Hyderabad, ICSE Companion.

Al-Linjawi, A. A. & Al-Nuaim, H. A., 2010. Using Ailce to Teach Novice Programmers OOP Concepts. JKAU:Sci, 22(1), pp. 59-68.

Armoni, M., 2013. On Teaching Abstraction in Computer Science to Novices. Jl. of Computers in Mathematics and Science Teaching, 32(3), pp. 265-284.

Armstrong, D. J., 2011. The Quarks of Object-Oriented Development. A survey of nearly 40 years of computing literature which identified a number of fundamental concepts found in the large majority of definitions of OOP, in descending order of popularity: Inheritance, Object, Class, Encapsulation, Method, Messag Passing.

Avraamidou, A. & Monaghan, J., 2009. ABSTRACTION THROUGH GAME PLAY. Thessaloniki, Proceedings of the 33rd Conference of the International International Group for the Psychology of Math.

Bezzola, L., Merillat, S. & Jancke, L., 2012. Motor Training-Induced Neuroplasticity. GeroPsych, 25(4), pp. 189-197.

Cankaya, I. A., Yuksel, A. S., Arif, K. & Yigit, T., 2015. A Dynamic Content Generation Tool for OOP Course. International Journal of Computer Theory and Engineering, 7(1), pp. 66-69.

Claassen, N., du Toit, L., Hoffman, D. & Vosloo, H., 1987. Manual for the General Scholastic Aptitude Test. Pretoria: Instute for Psychololgical and Edumetric Research.

Coolidge, F. L. & Overmann, K. A., 2012. Numerosity, Abstraction, and the Emergence of Symbolic Thinking. Current Anthropology, 53(2), pp. 204-225.
Farrell, J., 2010. Microsoft Visula C# 2010: An Introduction to Object-Oriented Programming. 4th ed. Canada: Course Technology, Cengage Learning.

Hamilton, R., Messing, S. & Chatterjee, A., 2011. Rethinking the thinking cap : Ethics of neural enhancement using noninvasive brain stimulation. Neurology, 76(187), pp. 187-193.

Klingberg, T., 2010. Training and plasticity of working memory. Trends in Cognitive Sciences, 14(7), pp. 317-324.

Mudhovozi, P., 2012. Social and Academic Adjustment of First-Year University Students. Journal of Social Science, 33(2), pp. 251-259.

Perwej, Y. & Parwej, F., 2012. A Neuroplasticity (Brain Plasticity) Approach to Use in Artificial Neural Network. International Journal of Scientific & Engineering Research, 3(6), pp. 1-9.

Rogerson, C. & Scott, E., 2010. The Fear Factor: How It Affects Students Learning to Program in a Tertiary Environment. Information Technology Education, Volume 9, pp. 148-170.

Rute-Perez, S. et al., 2014. Challenges in software applications for the cognitive evaluation and stimulation of the elderly. Journal of NeuroEngineering and Rehabilitation, 11(18), pp. 1-10.

Sinaceur, B. H., 2014. Facets and Levels of Mathematical Abstraction. Standards of Rigor in Mathematical Practice, 18(1), pp. 82-112.

Varma, S. & Schwartz, D., 2011. The mental representation of integers: An abstract-to-concrete shift in the understanding of mathematical concepts. Cognition, Volume 121, pp. 363-385.

Zhang, X., 2010. Assessing Students' Structured Programming Skills with Java: The "Blue, Berry, and Blueberry" Assignment. Information Technology Education, Volume 9, pp. 228-235.