Hosted by

Department of Computer Science
University of Botswana

Programme Committee

Nyongesa H. (Chairman)
Francis. J. Oguwu (Conference Chair)
Sampson D. Asare
Talib M.
Ojo S. O.
Masizana A.
Auwah E. B.
Anderson G.
Ayalew Y.
Gopolang B.
Eyitayo A. O.

Sponsors

Department of Information Technology

D C D M CONSULTING, Gaborone, Botswana

University of Botswana

ISBN 99912-572-4-1

Copyright © Department of Computer Science 2005
University of Botswana
Private Bag 00704
Gaborone
Botswana
# TABLE OF CONTENTS

Striving for IS III Project Excellence: the Views of Graduates. .............................................................. 1  
*Tom Addison*

What do we achieve by making honours students write research papers? ................................................. 5  
*P M Alexander and J H Kroeze*

Experience with Porting a Benchmark Suite to QNX .................................................................................. 11  
*G. Anderson and A.O. Eyitayo*

Usage of a WebCT-based Course in the University of Botswana: A Case Study from GEC121 ...................... 17  
*G. Anderson and T. Mogotlhwane*

Using Self-Organizing Maps to Analyze First Year IT Results in Relation to the Student’s Matriculation Results .................................................................................................................................................. 22  
*J Anniroot†*, EJ *Dean*

Impact of Intranets on Executive Information Systems Implementation in Organisations in Kwazulu-Natal 28  
*Udo Richard Averweg*

ICT considerations for business organisations’ information needs in a Metropolitan Municipality in South Africa .......................................................................................................................................................... 37  
*Udo Richard Averweg*

Routing Model for Computer Network ......................................................................................................... 43  
*E. B. Awuah, F. J. Ogwu, S. D. Asare and G. A. Aderoumu,*

Academic Literacy in Information Systems Students ..................................................................................... 50  
*Susan Benvenuti*

Towards identifying predictors of student performance in learning computer programming at a South African University ......................................................................................................................................................... 57  
*C Blewet and, D Achmad*

Some human factor issues in an e-learning enabled tertiary institution ......................................................... 64  
*Samuel Chikasha  Nephas Mufutumari and .Wim Van Petegem*

Performance of web-based 2-tier middleware platforms .............................................................................. 69  
*Johnson O. Dehinbo*

UNIX 4U2: Why I use Linux on my desktop ................................................................................................. 75  
*Rob Dempster*

The importance of teaching creativity and innovation to first year Systems Design Students. ..................... 80  
*S Dimas*

Intelligent Agent Driven E-learning Content Delivery in 3G Networks. ....................................................... 85  
*Joseph Bhekizwe Dlodlo ad Bakhe M Nleya*

Problem Solving and Object-Oriented Program Design: a Pedagogical Approach ..................................... 90  
*Adekunle O. Eyitayo*

Using VLANS to revolutionalize the organizational structure ........................................................................ 96  
*G. Mosweunyane, T. Zuva and K.G. Dibetso*

Factors influencing Students’ perception of e-learning in undergraduate basic level Computer course at the University of Botswana. ........................................................................................................... 102  
*Daniela Giannini-Gachago and Deepti Garg*
Roller coaster riding: highs and lows of understanding OO .......................................................... 109
Leila Goosen and Vreda Pieterse

Influence of Local Area Network Technology Advances in Local Area Network Design .................. 115
B. Gopolang and T. Zuva

Academic Performance of Postgraduate Assistants...................................................................... 120
Ken Halland, Lazarus Aron and Ruth de Villiers

Closing the distance when teaching a programming language .................................................. 127
D van Heerden

A Community Outreach Programme for Information Systems Students at the University of Cape Town --- 133
Jean-Paul Van Belle and Derek Smith

Software patents in the context of Southern African countries.................................................... 138
Bob Jolliffe

Implementing Information Security Management through Five Non-Technical Pillars................. 144
E. Kritzinger and S.H. von Solms

IT Service delivery and service support in South African Universities: The characteristics and pitfalls of a generic approach ................................................................. 150
AC Leonard and M Wheeler

Changing the Dynamics of Group Work: The Survivor Twist Case Study at the University of KwaZulu-Natal, Pietermaritzburg Campus ............................................................. 155
Lara Main-Baillie

Staff views on the use of WebCT in teaching a computer literacy course: a case study of the University of Botswana ........................................................................................................ 158
Keneilwe Mapoka and Oduronke Eyitayo

Selecting a Qualitative Research Approach for Information Systems Research ......................... 163
Alta van der Merwe, Paula Kotzé and Johannes Cronje

Reconsidering the relevance of IS curricula for Southern Africa.................................................. 168
Willem P. van der Merwe

Empirical computing costs in a computing training environment .............................................. 174
Opelo Mogothlwane

Assessing the perception of Undergraduate Students of the South African Military Academy towards and the value of ICT/Computers .............................................................................. 177
W. Moswetsi, J.C. Renken and A. Neethling

Dynamic load balancing techniques on Heterogeneous Network of Workstations: A survey .......... 182
Nkwebi P. Matlogelwa and Conrad Muller

ICT and the Digital Divide at the University of Limpopo: so what’s happening? ......................... 189
Naidoo, D, Scholtz, A and Cetwayo, N

Students Cheating in Assignments: How does this affect Examination Results? ....................... 193
Elsa Naude and Tertia Hörne

Bridging the Digital Divide in Primary and Secondary Education in Matabeleland North Province of Zimbabwe: Improving Education through ICTs ........................................ 198
Nomusa Dlodlo

Application of Blogging as a Tool Publishing Student and Teacher Portfolios .......................... 202
Nomusa Dlodlo
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Towards developing and applying a First Year Information Systems Curriculum on a Multi-Campus University</td>
<td>210</td>
</tr>
<tr>
<td>Norman H. Buller and Rosemary D. Quilling</td>
<td></td>
</tr>
<tr>
<td>Models for Optimisation of the Effectiveness of Information Retrieval Systems.</td>
<td>217</td>
</tr>
<tr>
<td>H. Okola Nyongesa, and Bontle Gopolang</td>
<td></td>
</tr>
<tr>
<td>Key Issues emerging from the Computing Curricula Development trends and Issues for University Computing Curricula Provision.</td>
<td>221</td>
</tr>
<tr>
<td>Ojo S.O., Masizana A. and Mpho T.</td>
<td></td>
</tr>
<tr>
<td>The next challenge: Aspect-Oriented Programming</td>
<td>230</td>
</tr>
<tr>
<td>K Padayachee and JHP Eloff</td>
<td></td>
</tr>
<tr>
<td>Factors to consider when forming Software Engineering teams in a final year module</td>
<td>234</td>
</tr>
<tr>
<td>Vreda Pieterse and Linda Marshall</td>
<td></td>
</tr>
<tr>
<td>Improving information systems students’ understanding of business processes</td>
<td>240</td>
</tr>
<tr>
<td>S. R. Ponelis and P. Joubert</td>
<td></td>
</tr>
<tr>
<td>A Project Web Site – Using WebCT to facilitate peer review in IS teaching</td>
<td>244</td>
</tr>
<tr>
<td>GYS LE ROUX</td>
<td></td>
</tr>
<tr>
<td>Methods for evaluating Interface Designs within an undergraduate systems development project</td>
<td>248</td>
</tr>
<tr>
<td>Suzanne Sackstein and Sasen Benvenuti</td>
<td></td>
</tr>
<tr>
<td>The next generation Web Technologies and their impact on e-learning</td>
<td>252</td>
</tr>
<tr>
<td>Manish Shrivastava and Alka Shrivastava</td>
<td></td>
</tr>
<tr>
<td>My marriage to cousin Lily</td>
<td>257</td>
</tr>
<tr>
<td>Shavren Singh and Corné de Ridder</td>
<td></td>
</tr>
<tr>
<td>The impact of ICT in Community based Learning</td>
<td>265</td>
</tr>
<tr>
<td>Ndabezinhle Soganile</td>
<td></td>
</tr>
<tr>
<td>Matric maths as a predictor of success in Information Systems – a study of Information Systems students at the University of the Witwatersrand.</td>
<td>268</td>
</tr>
<tr>
<td>Linda Spark</td>
<td></td>
</tr>
<tr>
<td>$\text{S-BASIC}^\text{®}$ in Education</td>
<td>272</td>
</tr>
<tr>
<td>$^1$R. Charakupa, Y. Liu, D.M.Totev.; S. D. Asare and E. B. Awuah,</td>
<td></td>
</tr>
<tr>
<td>The development of a Plagiarism Detection System</td>
<td>276</td>
</tr>
<tr>
<td>Lynette van Zijl and McElory Hoffmann</td>
<td></td>
</tr>
<tr>
<td>Local security, user and network access policies a cost effective way to securing a computer system.</td>
<td>281</td>
</tr>
<tr>
<td>Mr. Mthulisi Velempini</td>
<td></td>
</tr>
<tr>
<td>Integrity, Confidentiality and Security of Data and Information in a Networked Environment, a Challenge to all Educational Institutions in Maintaining their Integrity.</td>
<td>286</td>
</tr>
<tr>
<td>Mthulisi Velempini</td>
<td></td>
</tr>
<tr>
<td>Word processing meets Typography</td>
<td>290</td>
</tr>
<tr>
<td>Prof Lucas Venter</td>
<td></td>
</tr>
</tbody>
</table>
Roller coaster riding: highs and lows of understanding OO

Leila Goosen
Department of Science, Mathematics and Technology Education
University of Pretoria
Email: lgoosen@gk.up.ac.za

Vreda Pieterse
Department of Computer Science
University of Pretoria
Email: vpieterse@cs.up.ac.za

Abstract
This study was undertaken in order to determine how difficult (or not) first year computer science students perceive understanding basic object-oriented (OO) concepts, how well they actually understand these concepts and to identify the reasons why they find the acquisition of certain OO concepts to be more difficult than others. By analysing this information and ideas of best practices provided in related literature, suggestions are formulated for improving instruction of OO concepts.

Keywords
Object-Oriented, Pedagogy, Teaching OO concepts

1. INTRODUCTION

Students' acquisition of basic object-oriented (OO) concepts appears to be a major source of difficulty [3], because understanding these concepts requires them to be familiar with some potentially troublesome terminology [11]. Knowing which OO concepts students find difficult to understand can allow lecturers to structure instruction in such a way that understanding is improved [15].

The data collection mechanism was relatively informal and the results are not surprising for an experienced computer science educator. However the magnitude of discrepancies between students' perceptions of how well they understand the OO concepts and their actual understanding of the concepts was unexpected. This can be attributed to misconceptions held by students, which need to be addressed.

2. SCENARIO

Data for this study was obtained from first year students enrolled for Computer Science or Computer Engineering degree at the University of Pretoria (UP) during the second semester of 2004. Prior to the survey, 18 lectures covering the concepts of OO programming, as well as six tutorial classes explaining these concepts by means of practical examples, were offered to the participants. The survey instrument used was a questionnaire intended to capture subjective data about students' perception of the difficulty of understanding basic OO concepts, including inheritance, encapsulation and polymorphism. Enquiry was also made into the reasons why they find the acquisition of certain OO concepts to be more difficult than others. Finally, they were asked to supply explanations of the various concepts tested, in order to gauge their actual understanding of these concepts.

Participation was voluntary, with participants being offered a small number of bonus marks for participating. The students were allowed to participate in the online survey at their own convenience during a 72 hour period. From the 381 enrolled students, 245 (64%) usable answer sets were obtained.

3. STUDENTS' PERCEPTIONS

Difficulty Of Understanding OO Concepts

Subjects were asked to rate their perceived difficulty of understanding the following concepts on a four-point scale.
- objects and classes
- state and behaviour,
- encapsulation,
- inheritance
- polymorphism.

The options available for students to choose from, as well as the number of students who selected each of the options for the different concepts, are presented in figure 1. Objects and classes, and inheritance show very similar profiles in terms of distribution across the given options, with more than 80% of students claiming that they found it easy or very easy to understand these concepts. Almost 60% of students also claim that they found the state and behaviour of objects easy or very easy to understand. However, in the case of encapsulation, this trend starts to show a definite reversal, in that as many students find it difficult to understand this concept as those who find it difficult to
understand. For polymorphism, there are 4% more students who find it difficult to understand the concept than there are students who find it easy to understand.

Easiest And Most Difficult OO Concepts

In order to better understand students’ ratings for the various concepts reported in the previous section, they were asked to specify which OO concepts were, respectively, the easiest and most difficult to understand, and, in each case, why they found this specific concept easy/difficult to understand. The number of times that the various concepts were identified as being the easiest / most difficult is presented in figure 2.

![Figure 1. Perceived difficulty of understanding OO concepts.](image1.png)

![Figure 2. Easiest and most difficult OO concepts to understand.](image2.png)
In contrast with the previous section, where objects and classes, and inheritance seem to be equally easy to understand, in this section, objects and classes are reported almost 20% more often as the easiest concept compared to inheritance. Although inheritance is considered to be the easiest concept to understand by almost a third of students, there are also 10% of students who consider this to be the most difficult concept to understand.

In terms of the most difficult OO concept for students to understand, results obtained here for polymorphism and encapsulation confirm the difficulty ratings reported before. Apart from the five basic concepts, other concepts mentioned by students as being easy to understand include re-use of code (24), overloading (2) and constructors (4). Four students feel that everything is easy, and three that nothing is easy. Other concepts mentioned as being difficult to understand include copy constructors, overloading (7) and interfaces (9), while ten students found nothing very difficult.

**Actual Understanding**

The criterion we used as measurement of the actual understanding of a concept was the students' ability to explain the concept by means of a short paragraph. Their answers were then rated according to the following scale:

1. Very vague / not applicable / wrong.
2. Some appropriate content, but imprecise.
3. Answers are right, but don't convince the reader that the student knows exactly what the concept is about.
4. Accurate and articulate; student seem to understand the concept very well.

After both authors worked together on an initial calibration batch, classifications of students' explanations were made by each author individually and finally matched. Results obtained are represented in figure 3.

The expected pattern would be that concepts perceived as difficult should result in low understanding being achieved [13]. In this study, only encapsulation and polymorphism conform to this norm. State and behaviour is the best-understood OO concept of those tested, while objects and classes, and inheritance show many more students who don't really understand these concepts than would be expected from the difficulty ratings assigned.

We are aware that students could have used their notes to supply answers that were supposed to reflect their actual level of understanding. The students were however given the benefit of the doubt. We accepted that a student could have remembered the text they had studied for a test that was taken five days prior to the survey. In spite of this relaxed approach when these answers were rated, the contrast between the student's actual understanding of the concepts, and their own perception of how easy these concepts are to understand, is immense.

### 4. STUMBLING BLOCKS AND STEPPING STONES TO UNDERSTANDING OO CONCEPTS

---

4 Number of students mentioning a particular concept.
In an attempt to identify the sources of difficulty for our students in understanding particular concepts, and thereby improve instruction to support students in better understanding, their explanations as to why they find some concepts easy and others difficult to understand were examined.

The following summarises the opinions that were expressed by many students about the reasons why the concepts are easy or difficult to understand. Our interpretation of these opinions is related to similar or opposite findings by other scholars. We conclude each topic with some suggestions on what can be done to improve the learning environment for those concepts perceived to be difficult, and also what we have to continue to underscore when teaching the concepts to maintain the understanding of the easy concepts, as effortlessly as possible.

4.1 Objects and Classes

It has been reported that accepting the idea of classes and objects is not only the most important, but also the most difficult one for students [1,7]. The majority of students in this study confirm the pivotal nature of OO understanding, as do others in a similar study [3]. Respondents in this study found objects and classes the easiest of the various OO concepts to understand, due to many real world associations. The fact that these fundamental concepts of OO were not considered difficult to understand may reflect a teaching emphasis given to these vital concepts [13]. The fact that 56% of the students have not been able to reveal that they have grasped the concept of classes and objects shows that some of the students in this study perceived the concept as easy while this perception is based on a false idea that they understand it while they don't. Classes and objects can, and should, be introduced early in a course, and used subsequently [15]. Students in our study felt that if the concept of an object is related to real objects in the real world, the concept can easily be grasped since students deal with objects in the real world all the time. The modelling of these real world objects using practical examples that make sense can enhance understanding. Using the idea of grouping of similar real world objects into classes can be applied to define the concept of a class as being the description of many objects belonging to the class. Emphasising the fact that the concepts of objects and classes are the building blocks of OO programming may motivate students to go to a more trouble to understand them.

These student opinions emphasises the importance of relating objects in programming to real world objects and to model real world objects in the programming examples that are used.

4.2 State and Behaviour

Only a small number of students perceived state and behaviour as being the easiest or most difficult (see figure 2). As a relatively small number of comments accompanied these statements, these concepts did not feature prominently in this study. It seems as if the students did not experience major problems in grasping these concepts and could easily reveal true understanding of the concepts of state and behaviour of objects. This topic is however closely related to the basic concept of objects and classes and should be emphasised similarly.

4.3 Encapsulation

As OO concepts require a higher level of abstraction [3], these ideas need to be presented at an appropriate level of abstraction [12]. It is of interest to note that some problems in this regard can actually be solved by using encapsulation to build a cleaner, higher level, and more easily used, abstraction. It is, however, important that the related advantages of encapsulation and information hiding be emphasized [4], as it is difficult to understand abstract concepts such as these [14]. Although the students' actual understanding of these concepts is found to be considerably better than that of other advanced concepts like polymorphism and inheritance, it is much worse than that of objects and classes. The perceived difficulty of understanding encapsulation correlates better with our measurement of actual difficulty of understanding the concept since almost the same number of students who revealed poor understanding of the concept, indicated that it is the most difficult to understand.

Many students in our study attributed their inability to understand the concept of encapsulation to general issues such as not having sufficient notes or poor examples. A few students were, however, able to identify the essence of why understanding of this concept seemed to be evasive: They articulated that the concept of encapsulation is not self-explanatory and can not easily be related to something that happens in the real world. Some found the definition that was given, to be complicated.

To our opinion a more straightforward definition of what is meant by encapsulation have to be introduced. The fact that encapsulation is an abstract concept that has practical advantages when used correctly, need be stressed. This may reverse the students' urge to be able to relate the concept to things that happen in the real world, and can motivate them to want to have a deeper understanding of the concept.

4.4 Inheritance
Lecturers should note that their attitudes help to determine whether students view inheritance as a difficult topic [12]. This is however only one side of the coin. In this study over 80% of the students rated the concept of inheritance as easy or very easy to understand, indicating that they possessed the correct attitude. In spite of this less that 25% revealed satisfactory understanding of the concept. A case is reported where, although inheritance and polymorphism were introduced conceptually and through examples, students did not use these features in their own programs [15]. Students in the current study also acknowledged the importance of supplying them with opportunities to practically implement their knowledge of OO concepts in order to enhance understanding. Instead of only having purely descriptive introductory material, "hands-on" programming exercises should be utilized [3].

It was articulated by the students in this study that using inheritance and polymorphism entails a more complex way of thinking about programming. They pointed out that there are difficult, often confusing, concepts and terminology to master, before they can start to use it. They revealed that they only understood it after these concepts were implemented in programming examples. A number of students found that the programming examples that were used, adequately assisted them to rapidly gain understanding especially because the concept could be explained using small pieces of code. Some students said that they found the concept easy to understand, but related the concepts to things in the real world like inheriting something from your parents, revealing some misconception of the concept. If we listen to what the students are saying we should realize that, like with encapsulation, the abstractness of the concept and the fact that it is a programming practice to maximize re-use of code, rather than a concept that can be related to something in the real world, should be voiced clearly. The use of small pieces of code that can both explain the concept and reveal the value of inheritance is indispensable.

4.5 Polymorphism

The concept of polymorphism is the only concept that was rated as difficult or very difficult to understand by more than 50% of the students. It also stands out at the topic that got the most votes for the most difficult of all the topics. This correlates well with the observation that only about 10% of the respondents could show their understanding of this concept. The difficulty of the concept of polymorphism can be attributed to the fact that it was perceived by many students as very abstract. Various answers by students emphasises the fact that understanding of this concept is virtually impossible if it is not demonstrated by means of practical coding examples and exercises. The students complained that the introduction of new keywords and the complex syntax contributed to the difficulty of understanding and applying the concept. Some confusion between overloading of methods and the concept of polymorphism was also detected. As with the previous abstract concepts, it is extremely important that the abstract nature of this concept is revealed to the students. It should be illustrated clearly that the use of interfaces that create polymorphic behaviour is an excellent programming technique to write code that is re-usable to a greater extent. It is also evident that the relation between overloading (early binding) and polymorphism (late binding) needs to be emphasised.

5. RECOMMENDATIONS

All the concepts should be introduced simply using clear and concise definitions. They should be illustrated using code examples modelling real world objects. Care should be taken to ensure that the code examples are realistic, not too complex and they illustrate the concepts and the purpose of their use. Traditional lecture methods should be supplemented to enhance student understanding of difficult concepts [5] by steadily replacing these by a learner-centred approach where students have more responsibility for their own learning [10]. Laboratory assignments should be tightly coupled with lectures, enabling experiential learning accompanied by lecturer guided discovery of required concepts, as these offer a particularly useful means for promoting student understanding [4,9], as one of our respondents stated "Lecturers shouldn’t only explain concepts, but should also give students more time and enough opportunities to really practice these concepts". In terms of mutual support, pair programming can also be used beneficially in the introduction and demonstration of more conceptual material [8].

A pervasive idea seems to be to present concepts in broad strokes first and add details later [11]. Although student interest can be ignited by encounters with stimulating problems [6], a start should be made by introducing concepts simply, with a lot of assessment to determine exactly what students are learning and missing [16]. The basic concepts of objects and classes, as well as the concepts of state and behaviour of objects, need to be introduced early and emphasised adequately throughout the course. Students first have to be allowed to master basic facts, features and rules and gain insight into how these relate to existing knowledge [14]. They find it easier to understand when they familiarize themselves with high-quality, clear