Wits School of Education
Division of Educational Information Technology
27 St Andrews Rd •. Parktown• Private Bag X3 • Wits 2050 • (011) 717- 1000

Thursday, 24 October 2013

Open letter to the Minister of the Department of Basic Education and the MEC of the Gauteng Department of Education

Pinning Our Hopes on Obsolete Information Technology Tools is Unwise

The recently published Circular S9/2013 by the Department of Basic Education (DBE) on the standardization of software tools in the implementation and accessing of Computer Applications Technology (CAT) and Information Technology (IT) in schools demonstrated short-sightedness in the national government. From our point of view the government should be pushing for teaching core principles first then followed by core practices. Developing the core principles of programming earlier enable learners to participate in IT beyond basic computational thinking. The DBE, if not careful with its fascination with 'teaching technologies' is running a risk of locking IT education in our schools (especially public schools) into very limited technologies. Access issues need to be a central consideration given our context. Delphi and MS Office offer limited access for poorer communities.

Therefore, pinning our hopes on Microsoft Office alone and Delphi above Java is unwise. If there are short comings in the current curriculum, there is no better time than this to rethink other aspects of the curriculum. We need a curriculum that will enable us as educators to produce learners that will participate in complex systems development as well as in high-performance computing. For these two reasons [complex systems development and high-performance computing] a new framework or model that exposes fundamental programming principles and the development of computational thinking skills at various levels is needed. Those national examinations should be based on the core principles of programming and in case of practical experience we should not be limited to inaccessible or outdated technologies. Learning the core programming principles enables learners to develop mental orientations to be able to deal with various abstractions at various levels of programming, be able to examine how well a solution is crafted and scales across different sizes of problems and then be able to develop optimal algorithms. The DBE should make sure that there is a well-developed body of knowledge and experimental methods for schools instead of limiting what technologies are to be used.

The national government needs to recognize the importance of accelerating development in the Information Technology field in order to catch up with our counterparts in the BRIC countries as well as within Africa and the rest of the world. Competitiveness in the field of Information Technology is crucial in the digital revolution and in our quest to drive innovation in education and the industrial sector. In addition competitiveness promote long-term economic growth, lifelong learning and improves the overall effectiveness and productivity of any society. The DBE cannot send our country back to the Stone Age by doing away with Java and retaining an out of date programming language like Delphi.

Today most thriving economies in the ICT have adopted open source as the way to go. By retaining only Microsoft and Delphi we are saying no to inclusivity and we have to now deal with issues of licensing and access. We hope that it is clear that there are no costs associated with Java IDE like there are for Microsoft products. As an Educational Information Technology (EDIT) division at the Wits School of Education we believe that an intellectual discourse should have preceded such a decision. The specialists in the subject should have been part of the conversation [if this did occur then the EDIT division of the University of the Witwatersrand was not involved], yet we are at an advantageous position as we have a systematic relationship with a network of schools and diverse communities both educational and industrial. The EDIT division has a vested interest in making sure that IT teachers and learners command the necessary programming skills that will ensure South Africa's skills shortage in computing systems, software development and networking is a thing of the past.

The current problem in South Africa is that computer science education / information technology is not taught as standard in schools (Ashley, 2012)[1]. Even in cases whereby computer science / information technology education is taught, the ICT curriculum heavily focuses on the front-end of computing. Therefore, we cannot expect South Africa to be a comparable center of ICT excellence with the current state of the curriculum (Ashley, 2012). Professor Kritzinger asserted that "in most cases…learners are not taught about the new and upcoming technologies" (p. 20). In India, computational science courses are prescribed in school from an entry age (Ashley, 2012). In the Singapore context the focus is on producing highly-competent multi computer program users from a very young age. In the US computer technology is also introduced at the kindergarten or elementary level (Ashley, 2012). According to Eric Schmidt, the Google Executive, "if you want to thrive in the digital future, you need people who understand all facets of its integration from the very beginning"(p. 19).

As a country we need to begin using new technologies to circumvent existing knowledge weaknesses. Java is a programming language that was created in the early 90s to represent the current ideas of software engineering. If our ICT in education is poorly conceived, ICT as an enabler for a full range of essential knowledge and skills in society will be threatened. The marginalization of higher-level programming languages like Java in our schools is a serious mistake. We are living in a highly complex society and the need for complex systems or dependable software-intensive systems is rapidly growing. In order to create them we need to advance our

[1] Ashley, A. (2012). Are Computer Lessons Out of Date? *ITWeb Corporate IT Training Guide 2012.* Pgs 18 – 24.

IT curriculum so that it meets the most pressing demands of complex information systems. We need to learn to leverage rich emerging platforms that enable fundamental knowledge in programming abstractions and architecture.

Our objection to removing Java is based on our experience and what is going on in education and industry around the globe as well as in the software community. When thinking more about object-oriented programming development, Java gives an edge because it abstracts learners much more from the machine and that is an important part of development. For example, Delphi and C++ does not force learners to program in OOP style, while Java does that. It is advisable to use a language like Java that gets learners to program in an object-oriented style. In Java even the simplest program involves both methods and objects. The manner in which you introduce object-oriented concepts is clear and meaningful from the beginning.

It would be useful to be informed about the formal and rational assessment that was done on Delphi and Java before that decision was made to terminate Java and then we will then be able to engage more meaningfully in a debate. Currently we do not agree with the decision of phasing out Java and make Delphi an official programming or development language to teach Information Technology. Looking at Java, it is the most popular language in education and other industries so why not think about the long-term benefits of learners' development and retain Java rather than Delphi? In the Republic of South Africa we are dealing with issues of scarce technical skills yet we are making decisions which could further deepen the skills crisis. Our ability to address the country's needs in energy, health, transportation, etc. depends on our fundamental knowledge and experience with cutting edge information systems at various levels.

Delphi with its popularity in desktop application and native apps does not in any way claim dominance in that space. Java is the language of choice around the world. We need learners to function in multiple platforms instead of being Delphi developers which always demands an extra step for multiple platforms. Java is a cross-platform programming language compared to other programming languages. With Delphi and C++ there are considerable deployment issues because there are no Delphi and C++ virtual machines. We would even consider C# instead of Delphi as it is another almost pure object-oriented language. In our view, if the government wants to teach object-oriented programming then Java is the way to go. The theoretical basis in Java is much simpler than other languages except possibly C#.

Let us look at India and Singapore who are highly-competent multi computer program users - they are developed at a very young age and we will never compete with them if our curriculum keeps focusing on front-end of computing. IEB schools are always ahead of public schools because they respond to societal needs. We are convinced that they will never let such policies get in their way of developing learners who are highly competent in Information Technology. Those schools realize that we lack the fundamental knowledge needed to harness IT potential within a broader societal context. We need to look at our own country and think about the programming or technical challenges posed by future complex systems. Are we going to continue to depend on off-shore companies or we are going to leverage our own economic strength?

The emergence of programming technologies for parallel multi-core processors is calling for complex programming languages better than Delphi. We need to seize the moment now and develop an IT in education curriculum that will participate in the development and evolution of software-intensive systems. Those reading this need to understand that by no means we are saying in Java is easy to get things done, but the standard libraries, garbage collection and code refactoring tools do help a lot. For example Java or C# learners do not have to worry about garbage collection issues as it is automatic. In this case we are arguing that Java should be the language of choice as it eliminates the licensing issues, access and develops learners. Java is more intuitive than Delphi because learners do not have to deal with pointers and memory management. In this document we are not arguing for learners to align with one programming language only, but we are looking at it in terms of pedagogy and learning. The fact that Java code is written once then can run anywhere is good enough to be the language of choice in schools. So spending so many years studying something that is not used or popular in education and industry will not be beneficial to the learners. We are aware that languages come and go but Delphi in all honesty has outlived its time. We need to teach languages that are sufficiently powerful to teach OOP techniques with simple syntax. The pragmatic approach should be to teach a language that is widely used in industry and there is no doubt that Java is one of the languages not Delphi.

Java can be used for various desktop apps, web apps, server apps, etc. This is not to say Delphi cannot do desktop apps; the problem is that it does not scale well in distributed environment. The memory issue is another issue and we are trying to manage our resources well. In contrast Java is built for distributed environment and is highly scalable. I am sure the DBE saw some value in Delphi and Microsoft products, but open source is the way to go given our context. Finally, the Java Virtual Machine can theoretically and practically compile many languages and Java is highly distributed, scalable, have real world use, up to date features, a lot of standard libraries written for it and is a high level language with rapid development principles.

From the Wits School of Education: Educational Information Technology Division
Dr. Reuben Dlamini (Head of Division)
Dr. Ian Moll
Ms. Sithabile Ndlovu
Mr. Tom Waspe