Technical Review in Response to “IMPLEMENTATION PLAN FOR STANDARDISING PROGRAMMING LANGUAGE FOR INFORMATION TECHNOLOGY FOR 2015”

In response to the recent criticism in the media, the Department of Basic Education responded with the document entitled "**IMPLEMENTATION PLAN FOR STANDARDISING PROGRAMMING LANGUAGE FOR INFORMATION TECHNOLOGY FOR 2015"**. Purpose of this technical review is to highlight some of the key technical inaccuracies found within the “standardisation document” that we believe has had a significant effect on the final proposal that supports Delphi as the programming language of choice over Java.

From this point on, the “standardisation document” shall refer to “IMPLEMENTATION PLAN FOR STANDARDISING PROGRAMMING LANGUAGE FOR INFORMATION TECHNOLOGY FOR 2015” and “Java” shall refer to Java the Programming language and the Technology unless explicitly stated.

“The complexity of translating abstract concepts into practical implementation is much reduced with Delphi syntax, inherently stable runtime environment and the simplicity of dependencies, reduced complexity and speed of compilation”

On page 2, Section “1. Problem Statement”, the following line reads, “The complexity of translating abstract concepts into practical implementation is much reduced with Delphi syntax, inherently stable runtime environment and the simplicity of dependencies, reduced complexity and speed of compilation”.

Stating that Delphi has an inherently stable runtime environment and then in the following sentence in the standardisation document state that Delphi is much less suited to complex, large-scale highly scalable solutions brings out an interesting fact in Java’s favour that has been omitted. This fact, and the one that the author(s) of the standardisation document acknowledge, is that Java runtime environment in terms of stability is suitable for complex, large-scale highly scalable solutions. Since this is the case there is no doubt then that Java runtime environment is more than a stable environment for a simpler setting such as the classroom.

With regards to the point on simplicity of dependencies, the context is the classroom setting. This is an important fact, as students will not be building large-scale systems that tend to have larger dependencies on other libraries. Thus simplicity of dependencies in this context is largely inherent and requires little further argument.

Inherent or intrinsic complexity of the Java Programming Language

In a number of sections of the standardisation document, it is stated that the Java language is inherently or intrinsically complex, but with insufficient justification to make this conclusion in the context of learning the fundamental concepts of programming.

Java being ranked as one of the most popular programming languages in the world [1] today, in both the industry and the academic setting, it is fair to say that Java has accumulated some complexity over time to suite the different programming paradigms and constructs relevant in the modern times we leave in. Striking the right balance between relevance and simplicity is the key here and we believe that Java strikes this balance more appropriately than Delphi does.

"Literature suggests that programming is hard to learn and that the choice of the environment and tool used for teaching is therefore very important"

On page 8, Section "3. Discussion", the following line reads, "Literature suggests that programming is hard to learn and that the choice of the environment and tool used for teaching is therefore very important".

Reference is made to (Kurland et al., 1989) and (Soloway, E. & Spohrer, J. (1989)). Both of these references date back to 1989. High-level programming languages such as Java, Delphi, Python, Ruby and Haskell become far more prevalent from the early 90s onwards [2]. This was the birth of the Internet age. Prior to 90s, programs were predominantly written in lower-level programming languages.

Technology has come a long way. Even our mobile phones today have enough computation power to run substantially complex programs written in high-level programming languages that are a lot easier to learn than the lower-level languages such as C [3].

Therefore, our objection here is that these references are outdated and no longer applicable in the current setting.

"Delphi is ideal for learning programming as it is a strict, yet forgiving language"

On page 13, Section "4. Recommendation", the following line reads, "Delphi is ideal for learning programming as it is a strict, yet forgiving language".

Java is a strict programming language [4] in that all formal parameters of any method/function have to be evaluated prior to evaluation of the method/function in question. Additionally, Java is a statically typed language [5] (compile-time type error checking), yet it is forgiving in the appropriately sound cases such as the support for type inference [6] and the support for generic types (parametric polymorphism).

By this definition, Java is "at least" equally ideal for learning programming.

“Compiler tells you where your errors are”

On page 13, Section “4. Recommendation”, the following line reads, “You don't need to worry about things such as case, its compiler tells you where your errors are”.

Java is a statically typed programming language [5], which means that the type error checking takes place at compile-time, in effect, the compiler tells you where the errors are.

The quote above mentions that you don’t have to worry about things such as “case”. The paragraphs to follow will apply if and only if “case” in the quote refers to the support for “case insensitivity” in Delphi programming language.

Computers are not forgiving. An upper case character holds different meaning to the lower case variant. They are entirely different and this should be taught as part of fundamental computing and programming.

On a contrary, Java is case sensitive, thereby explicitly stating the difference between the two cases (i.e. upper case and lower case).

Majority of programming languages largely in use today are case sensitive [7]. Teaching the students in school to practice case-insensitivity and then later, in higher educational setting and industry, to switch to case-sensitivity is illogical.

“New versions of Delphi offer Mac, iOS and Android compilers from the same IDE, and same language!”

On page 14, Section “4. Recommendation”, the following line reads, “New versions of Delphi offer Mac, iOS and Android compilers from the same IDE, and same language!”.

Firstly, this quote seems to have important emphasis, considering the exclamation mark.
In Java’s favour, what makes Java technology well recognized was captured well in its slogan, “Write once, run anywhere”.
Java compiles to byte code that is portable across computer architectures. Whereas, languages such as Delphi have to target a specific architecture type at compile time.

Secondly, it’s worth brining to attention here that in Delphi, there is no complete automated memory management. The programmer has to free object/references explicitly on many occasions and she/he has to keep a mental note of when to free up what. In times when high-level programming languages such as Java, are most widely use, this is a very ceremonious process and an unnecessary one to say the least. Modern programming languages typically employ garbage collection [8] to simplify memory management.

This is a very important omission in the standardisation document that undermines one of the key essential benefits of utilizing high-level programming languages in the first place.

Finally, the rollout plan has indicated the deployment of Delphi 2010 Academic Edition [9]. This version is free for schools for a limited period, which is not the new version of Delphi. Therefore making the statement in the quote non-applicable, for the time being, as Delphi 2010 only supports Windows.

"Several websites and blogs are available that support Delphi"

On page 15, Section "5.1. Benefits", the following line reads, "Several websites and blogs are available that support Delphi".

Firstly, it is important to restate that this statement falls under the "Benefits / Resources" section.

Stating that several websites and blogs are available that support Delphi should be a concern rather than a benefit.

Number of websites and blogs dedicated to Java on the Internet are far beyond several [10][11]. Java is an open technology that is a product of well-defined standards that are governed by the standards body called the Java Community Process [12] to which thousands of users, businesses and institutions contribute. It is not some closed technology built by one vendor; it's a technology that has been thriving on open community for almost two decades.

Structured Programming

On page 23, Section "Annexure B", under row two, titled "Structured Programming", Java is listed as "No (Not without utilising techniques to bend/side-step the intended purpose of the language)".

This is inaccurate.
By definition [13], structured programming is a programming paradigm aimed on improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures and for and while loops. Subroutine is a sequence of program instructions that perform a specific task, packaged as a unit. In different programming languages a subroutine may be called a procedure, a function, a routine, a method, or a subprogram. [14]

Then by definition, Java is a fully structured programming language.
A quick mapping for the uninitiated; subroutines in Java are called methods (abstractions of sequence of program instructions; static and instance based). Block structures are block statements in Java. For and while loops and a number of other iterative constructs are available in Java.

Accessing a database through programming language constructs

On page 26, Section “Annexure B”, regarding, “Accessing a database through programming language constructs & Setup a connection or connect to a database (single table) by providing path in code statements”.

Referring to the comment, “Knowledge of exceptions is required in the establishment of a connection object”.

Since a database connection is an IO operation, then what can one expect to happen if a connection to the database fails at runtime? Java answers this question explicitly by expecting the programmer to either acknowledge that an exception could occur (throws clause) or by providing an alternate path (try/catch) in case the connection fails. Just as the student needs to learn what data types are, she/he should learn what an exception type is. It’s a fundamental part of programming.

Query a database (single table) using simple SQL constructs

On page 26, Section “Annexure B”, regarding, “Query a database (single table) using simple SQL constructs”, under Delphi/Pascal it reads, “The process is easily accomplished by assigning a SQL statement to either a TQuery or TDataset (related) component.”

Complexity of 2 has been assigned to the above. Whereas under the Java column, complexity of 4 is assigned with one of the comments being, “Knowledge of the methods of the result set object is required in order to allow for functionality”. However, the same holds for TDataset type [15] in case of Delphi/Pascal, i.e. knowledge of the methods of the TDatase is required.

A general note is in order here regarding the complexity scoring mechanism. We question the logic behind scoring, as there seem to be rather large unjustified variances.

Comments from senior Java Developers in industry

On page 31, Section “Annexure B”, regarding, “Comments from senior Java Developers in industry”.

One of the senior developers states under “Java vs Delphi“ the following, “Delphi is syntax and environment stable and Java is not …”. This is an inaccurate assessment on many accounts.

Please refer to the answers in the above sections *“The complexity of translating abstract concepts into practical implementation is much reduced with Delphi syntax, inherently stable runtime environment and the simplicity of dependencies, reduced complexity and speed of compilation”* and *“Delphi is ideal for learning programming as it is a strict, yet forgiving language”*.

Summary of aspects that could be considered

On page 34, Section “Annexure C”, regarding, “Java requires Netbeans – not integrated but a bolt-on – could be problematic especially if correct versions/combinations are not used”.

This is inaccurate.

Firstly, it is trivial to obtain the Netbeans IDE with the Java language (JDK) as a single install [16].

Secondly, use of the word “bolt-on” here is discrediting to the Java IDE in question. IDE is NOT a programming language; it’s an integrated development environment. A “bolt-on” in technical term has a meaning of “an after thought”. That is not the case here. Programming language and IDEs are not one and the same.

Technical Issues

On page 35, Section “Annexure C”, regarding, “Technical Issues” under the Java column, “As Java is open source, stability could be questioned.”

Inaccurate generalisation. Java is based on a well-defined open standard. This does not mean that one has to use the open source version such as OpenJDK. Multiple vendors, such as Oracle, IBM and Apple, implement the Java language and offer the implementation for free use. These vendors sell numerous flagship products of which many are mission critical systems that run on these Java implementations.

Other

Numerous other irregularities exist that penalise Java in the standardisation document. For instance, under the “Cost” column, on page 35, consideration is “cost” (nothing more nothing less). Java and Netbeans (and most other Java IDEs for that matter) are completely free; multiple other non-applicable comments are made in the Java column in an attempt, to what only can seem to be to, devalue its strengths.

Notes

[1] - TIOBE Programming Community index is an indicator of the popularity of programming languages
http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

[2] - The 1990s: the Internet age
http://en.wikipedia.org/wiki/History_of_programming_languages#The_1990s:_the_Internet_age

[3] - Lower-level languages are relatively difficult to learn because a detailed knowledge of the internal working of the computer is required
http://c2.com/cgi/wiki?LowLevelLanguage

[4] - “Nearly all programming languages in common use today are strict. Examples include C#, Java, Perl (through version 5), Python, Ruby, Common Lisp, and ML.” http://en.wikipedia.org/wiki/Strict_programming_language

[5] - “Static type-checking is the process of verifying the type safety of a program based on analysis of a program's source code.”
http://en.wikipedia.org/wiki/Type_system#Static_type-checking

[6] - Type inference is a compiler's ability to look at each method invocation and corresponding declaration to determine the type argument (or arguments) that makes the invocation applicable.
http://docs.oracle.com/javase/tutorial/java/generics/genTypeInference.html

[7] - Some computer languages that are case-sensitive for their identifiers: Java, C++, C#, C, Ruby, Haskell, Scala, and XML.
http://en.wikipedia.org/wiki/Case_sensitivity

[8] - “Generally speaking, higher-level programming languages are more likely to have garbage collection as a standard feature.”
http://en.wikipedia.org/wiki/Garbage_collection_(computer_science)

[9] - Embarcadero Delphi 2010
http://en.wikipedia.org/wiki/Embarcadero_Delphi#Embarcadero_Years_.282008-.29

[10] - Top 20 Java Websites http://www.mkyong.com/featured/top-20-java-websites/

[11] - GitHub is a web-based hosting service for software development projects.
https://github.com/
GitHub Hosts over 157,618 Java projects and 901 Delphi projects.
http://adambard.com/blog/top-github-languages-for-2013-so-far/

[12] - The Java Community Process (JCP), established in 1998, is a formalized mechanism that allows interested parties to develop standard technical specifications for Java technology.
http://en.wikipedia.org/wiki/Java_Community_Process

[13] - Structured Programming
http://en.wikipedia.org/wiki/Structured_programming

[14] - Subroutine http://en.wikipedia.org/wiki/Subroutine

[15] - Data.DB.TDataSet list of methods
http://docwiki.embarcadero.com/Libraries/XE3/en/Data.DB.TDataSet_Methods

[16] - Java Development Kit (Compiler and runtime) with NetBeans
http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html