Determining Suitable Programming Language(s) for the B Tech (IT) Degree

JO Dehinbo

Abstract

There are various programming languages such as Basic, Fortran, Pascal, Cobol, C, C++, Visual Basic (VB) and Java, which are being taught within the Information Technology curriculum of tertiary institutions in South Africa. Due to differing features as in normal languages, some programming languages which are not easier to learn may lead to high failure rate. Some that are simpler to learn may not offer the flexibilities required for future works. The current practice of teaching many of these languages leads to repetition. Adequate consideration need to be given to the choice of a programming language that will be easy to learn even for first year students, and yet lead to increased productivity in future tasks.

The study involves evaluating four programming languages namely C++, Visual Basic, Java and Pascal, in terms of their ease of learning, ease of use under pressure, Line of Codes (LOC) and overall rating. Survey and Experimental approaches were used. Questionnaires were administered to respondents using any of the four languages: C++, Java, Visual Basic, and other structured languages (like Pascal or Basic) in different study groups. In addition, a simple programming exercise was given to respondents to solve and the estimated Lines of Codes (LOC) for the solution in each programming language was measured.

The study concluded that unlike Pascal, there is no significant difference in the factors studied for C++, Visual Basic and Java. The findings therefore show that a careful combination of the languages can achieve the desired result.

Keywords

Programming, languages, ICT, teaching, learning

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

1. Introduction

Various programming languages such as Basic, Fortran, Pascal, Cobol, C, C++, and Visual Basic have been used in the past as the programming language of choice for the various computer-based instructional offering for beginners in Tertiary institutions. In the Soshanguve Campus of the Tshwane University of Technology (formally Technikon Northern Gauteng - TNG), Pascal was used in 1999, C was used in 2000, and C++ has been used since 2001. The pass rate remains less that 50%.

1.1 The Context of the Problem

The use of various programming languages in computer-based training led to situations which necessitate some considerations. Adequate consideration will lead to the choice of a programming language that will lead to increased productivity. On the other hand, inadequate consideration could make programming problematic now and even in future. Below are some of the situations:

Basic is considered as a very simple language for beginners but cannot be used for complex programming. So after the first year, the students have to start with another programming language, which may lead to confusion of the statements in both languages. The new language may also not be mastered to the point where it can be used effectively for industrial work and research. Similar arguments apply for FORTRAN and Pascal.

Basic, FORTRAN, Pascal and C are structured programming languages that need a complete re-orientation to adjust to the more recent object-oriented programming style portrayed by C++, Java and event-oriented style of Delphi and Visual Basic. C++ is a hybrid language capable of the structured programming style in C as well as the object-oriented programming style compatible with Java.

However, C++ is considered by some people as being difficult, saying it seems too complex for first year students especially those with no prior knowledge of computing. Another consequence of the

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

complexity of C++ is that many postgraduate students do not enjoy using it for further complex programming thus abandoning it for Java.

Internet programming at present offers challenging opportunities for research especially in areas such as Servlets, CORBA, Java Server Pages, HTML, XML, COM/DCOM, and Enterprise Java Beans (EJB). These however need a good background in Java and not Basic, Fortran, Pascal or Cobol.

Many programmers, when given a choice of languages for a new project, continue to use the language with which they are most familiar, even if it is poorly suited to the new project. It is therefore very important for student to be trained using a programming language that will be most suitable for their future tasks.

1.2 The Problem Statement

The pass rate for technical programming language subjects for the past few years in TNG averaged less than 50%. This, in addition to various comments from students such as "C++ is too difficult" indicates that the various languages used so far, are possibly not easy to learn for beginners. There are some other languages which could be very good for beginners but do not have enough complexity for the student to become rigorous in programming at the end of the 4th year of a B Tech study. This leads to half-baked students with problems in securing employment and in undergoing postgraduate studies and conducting research.

1.3 The Aim of the Study

The aim of the study was to identify a suitable programming language for the new B Tech (Information Technology) degree in the Soshanguve campus of the Tshwane University of Technology and other tertiary institutions. This language should be relatively easy for beginners to write useful programs, to incorporate other current and relevant IT technologies, and broad enough to fill the 3 to 4 year curriculum while at the same time offer some prospects for future industrial use and research.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

1.4 Objectives of the Study

The objectives of the research are:

- Investigate the preferred or 'easy to learn' language among undergraduate students at different levels.
- Investigate the language that is considered most flexible.
- Investigate the language that is considered most suitable for complex works.
- Compute the Line of Codes (LOC) for a solution to a simple problem in each programming language.

2. Types of Programming Languages

There are basically hundreds of programming languages on the market today. According to Sebesta [7], it is widely believed that the depth at which we can think is influenced by the expressive power of the language in which we can communicate our thoughts. Those with a limited grasp of natural language are limited in their complexity of their thoughts, particularly in depth of abstraction. It is difficult for people to conceptualize structures that they cannot describe. Programmers in the process of developing software are similarly constrained. The language in which they develop software places limits on the kinds of control structures, data structures, and abstractions they can use; thus the form of algorithms they can construct are also limited.

Because of the great diversity in computer use, programming languages with very different goals have been developed. From widely accepted views in various programming texts such as [7], some of the areas of computer application and their associated languages are discussed below:

2.1 Scientific Applications

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010)

Typically, scientific applications have simple data structures but require large number of floating-point arithmetic computations. Efficiency is a primary concern. The first language for scientific applications was FORTRAN.

2.2 Business Applications

Business languages are characterized according to the needs of the application by elaborate input and output facilities and decimal data types. The first and the most successful high level language for business was COBOL which appeared in 1960.

2.3 Systems Programming Languages

The operating system and all of the programming support tools of a computer system are collectively known as its system software. A language for this domain must have low-level features that allow the software interfaces to external devices to be written. Examples of such language are Assembly Language and Extended ALGOL.

2.4 Very High-Level Languages (VHLLs)

The fourth-generation languages that were developed in the 1970s are sometimes considered to be VHLLs. These languages are used in the commercial or business area of computer applications. They have commands that are commonly programmed in languages close to English. Examples of such languages include DBASE IV, FOXPRO, and ORACLE.

2.5 General-Purpose Structured Languages

These are simple languages like BASIC, PASCAL and C that can be used for both scientific commercial applications. The central theme for these languages includes "top-down" problem decomposition and modularization. Modern programs are usually complex and being developed as a team effort. Top-down decomposition makes it easy to decompose the problem into manageable components or modules. Modularization makes it easy for different members of a team to work on different parts of the application being developed. These parts could be developed as modules or functions.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

This saves development time and enhances reuse of codes. Instead of rewriting the codes in the functions in different parts of the program, they are simply called to do the required task and return result values. Compile time is also saved because, no matter how many times the functions are called, they are compiled only once. Functions also improve maintainability, extensibility and reliability. This is achieved by allowing one to fix errors or bugs in one place, rather than every where a task is performed. Readability is thereby improved by isolating code that performs specific tasks [6], making it easier to maintain and extend the system, even in a reliable way.

2.6 Object-Oriented Programming Languages

According to Lerdorf & Tatroe [6], Object-oriented programming opens the door to cleaner designs, easier maintenance, and greater code reuse. Real life objects are used to model problem domain objects as illustrated by the following statements:

> "Unlike the procedure-oriented method of programming, the object-oriented method allows the programmer to use familiar objects to solve problems. The ability to use objects that model things found in the real world makes problem solving much easier [8]".

This object-oriented method covers basic topics such as Classes, Objects, Properties and methods, Encapsulation, Information hiding and Abstract data types, Inheritance, and Polymorphism.

Zak [8] defines a class as a pattern or blueprint used to create an object. When one designs the program, one has to decide the fields for each data item, and then come up with the functions that operate on those data items. In object-oriented terms, one is designing the class by so doing [8]. Most real life objects can be grouped into some classes based on their similar properties. For example, various types of cars can have properties such as "number_of_tyres = 4", "number_of_passengers = 5", etc.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

Recent examples of object-oriented programming languages used in educational institutions are C++ and Java. C++ is a hybrid language, combining the features of C with object-oriented features of Smalltalk. In Java, every program is a class, thus being completely object-oriented, though you can still use structured logic within each class.

2.7 Event-Driven Programming Languages

Event-driven programming languages include the likes of Visual Basic and Delphi. They are good for interface design involving such events as clicking a button to perform actions.

2.8 The Shift to Net-Centric Computing

The Internet and the World Wide Web are revolutionizing conventional business models and in some cases producing new ones [4]. Recently, most organizations started to adopt an approach of "write once, run anywhere", where programs are designed and installed on a web site and can be run from any computer connected to the internet. Central to this approach is the Java language, used in conjunction with middlewares such as Servlets, which can also access remote database systems. Dehinbo [3] observed that a very important feature that promotes the use of the Java language is the integration of Java to web pages through HTML.

According to Hamilton [5], Java is an object-oriented programming language with syntax similar to C and C++, only simpler. Because Java is an interpreted language, the typical C or C++ compile-link-load-test-debug cycle is reduced. The main attraction is the fact that Java applications are completely portable. Thus, you write your code once, and you never even need to port or recompile it. The Java runtime environment, or virtual machine translate the bytecode into actual machine specific instructions. Users are assured that applications are safe, even if downloaded from the internet, because the Java runtime environment, or virtual machine, has security mechanisms that protect against tampering.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

According to Hamilton [5], Java originated in early 1990 with James Gosling, a software developer at Sun Inc., who was part of a team investigating advanced software techniques for a variety of networked devices and embedded systems. The team's goal was to simplify the development of secure, high performance, and highly robust applications on multiple platforms in heterogeneous distributed networks. They first considered using C++, but because of the many difficulties encountered with C++, the team began to examine other languages, including Eiffel, Smalltalk and Cedar/Mesa. In the end they decided to develop an entire new language, drawing from the best features of each of these languages, with simplicity as one of the overriding design goals.

The designers of Java describe the language as C++ without guns, knives or clubs. They mean that the difficult parts of the language such as pointers and operator overloading that bedevil programmers, as well as learners, have been removed from the language itself and are implemented within the underlying layer [1].

2.9 Languages for Training Purposes

For training purposes in tertiary institutions, not all programming languages are suitable due to factors such as cost, complexity and availability of trainers. However, the four programming languages used in this study (C++, VB, Java and other structured programming languages like Pascal) serve as good representation of the most commonly used languages in educational institutions and industries today.

In a study conducted by Bergin [1] in 1996 to compare C++ with Java, it was concluded that Java is an improved version of C++, by eliminating some of the difficult concepts of C++ such as Pointers.

3. Research Methodology

3.1 Project Design

The approach that was used is a survey and experimental design. The stratified survey approach was used. For the survey part, the ques-

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

tionnaires contain closed questions, while for the experimental design, there were open questions whose answers are the program listing for the questions. Questionnaires were administered to respondents using any of the four languages: C++, Java, Visual Basic, and other structured languages (like Pascal or Basic) in different study groups. This is necessary in order to solicit information from a range of users of these programming languages.

The required information includes: ease of learning, ease of use under pressure, suitability for complex jobs, problems commonly encountered, programming language preferences, flexibility and efficiency.

For the experimental part, respondents were given a small problem to be solved using each of the programming languages; C++, Java, Visual Basic, and any other structured languages (like Pascal or Basic). The Line of Codes (LOC) for each solution in each language was measured.

3.2 Population

The population for the survey is all programmers and students using the four programming languages in their works and studies. Due to financial and time constraints, this was limited to respondents in software organizations within the Gauteng region as well as web lecturers, programmers, researchers in major higher institutions in the Gauteng region in South Africa.

At an estimated value of 3 programmers per organization, and 100 software organizations in the Gauteng region, we have about 300 subjects in software organizations. At an estimated value of 10 lecturers with 40 students per institution, and 10 higher institutions in the region, we have about 250 subjects in South African higher institutions. These give a total of about 800 subjects for the population.

The intended population is stratified into 8 groups from the Gauteng region, consisting of students currently using C++, Java, Visual

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

Basic, structured Programming languages like Pascal, BASIC or C, as well as experienced or Post Graduate users of these languages.

This is necessary in order to have an adequate representation of students and experienced users of the various programming languages.

3.3 Sampling Method

As stated by Corbetta [2], sampling is the procedure through which we pick out, from a set of units that make up the object of survey (the population), a limited number of cases (sample) chosen according to criteria that enable the results obtained by studying the sample to be extrapolated to the whole population. One approach to counter human error is to randomly split the participants into groups and treat all groups exactly the same in all other respects. The population is divided into strata and samples taken randomly.

According to Corbetta [2], for a 95% confidence level, a population size of about 800 requires a sample size of 300. However, given the scarcity and the busy schedules of programmers in industries, we can reduce the confidence level to about 80%. Therefore a sample size of about 100 would suffice for our population size of about 800.

3.4 Data Collection

A total of 160 questionnaires were distributed. Since each questionnaire contains questions pertaining to the four programming languages, if each questionnaire were fully completed, a total of 100 completed questionnaires are required. However, most respondents are familiar with 2 or 3 programming languages. In all, a total of 110 questionnaires were duly completed and returned.

3.5 Hypothesis

The null hypothesis is that the low pass rate for C++ is due to C++ being an exceptionally difficult and complex language, as there is likelihood of significant difference in the 'Ease of learning', 'Ease of use',

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010)

and 'Suitability for complex task' features for C++ as compared to other programming subjects such as Pascal, Visual Basic and Java.

3.6 Limitations

Due to the time frame and the cost of traveling, this study is limited to respondents from the Gauteng region.

4. Results

4.1 Ease of Learning

From the data in Table 1, twenty three percent of the respondents found Visual Basic very easy to learn. From usage experience in my personal view, this is likely to be true because VB has the "drag and drop program generating" facility in which you simply drag an object such as a Combo box unto the desktop, and then VB translate to equivalent program lines.

Next to VB, people found Pascal very easy to learn. This is true as Pascal has simple structured programming constructs.

Nine percent of respondents found C++ and Java similar in terms of ease of learning. This could be true as they have similar language constructs.

Table 1: Ease of learning the programming languages

	C++	Java	Visual Basic	Pascal
Very Easy	11	10	25	15
Relatively easy	10	45	35	5
Okay	54	20	35	25
Difficult	15	5	5	5
Very difficult	0	0	0	0
Don't know	10	20	10	50
No response	10	10	0	10

Forty one percent of the respondents found Java relatively easy. This could be explained due to the fact that Java is considered as "C++ without the pointer problem". Fifty percent of the respondents found C++ just okay. This fact is supported by the fact that for simple programs, C++ has few lines of code (LOC).

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

Fourteen percent of the respondents found C++ difficult. This is basically supported by the low previous pass rates at TNG which necessitated this study.

Forty five percent of the respondents don't know about Java. This is because Java is relatively new and only few programmers in "well-paid jobs" can afford specialized training in Java.

4.2 Ease of Use under Pressure

From the data in Table 2, thirty six percent of the respondents found Visual Basic very suitable for use under pressure. This is true because from experience, Visual Basic also has simple programming constructs as well as the "drag and drop program generating" facility in which you simply drag an object such as a Combo box unto the desktop, and then VD translate to equivalent program lines. This is followed by C++ which has fewer LOC than C++.

Table 2: Ease of use under pressure

	C++	Java	Visual Basic	Pascal
Not suitable	10	0	0	10
Manageable	10	25	10	10
Suitable	30	25	30	35
Good	15	30	20	10
Very suitable	25	20	40	0
No response	20	10	10	45

4.3 Suitability for Complex Jobs

From the data in Table 3, fifty percent of the respondents considered Java to be very suitable for complex jobs. This is true as Java is more flexible in linking to other current technologies such as the web-based tools like HTML and XML, and also web-based databases. This is the reason why there is increasing demand for Java programmers today. It is interesting to note that even 9% of respondents who gave no response on Java in terms of ease of learning above, still consider Java very suitable for complex jobs as they would at least have seen much advertisement for Java programmers.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010)

Table 3: Suitability for complex jobs

	C++	Java	Visual Basic	Pascal
Not suitable	5	0	0	10
Manageable	10	10	5	15
Suitable	35	20	30	20
Good	15	25	25	20
Very suitable	40	55	50	5
No response	5	0	0	35

4.4 Overall Rating for the Languages

From the data in Table 4, C++ and Visual Basic are considered to be equally best by 41% of the respondents, possibly due to its use in most schools today. Java comes next at 27%, probably due to the fact that it is newly being introduced in most schools.

Table 4: Overall rating for the programming languages

	C++	Java	Visual Basic	Pascal
Best	45	30	45	10
Better	15	25	10	0
Good	20	15	25	40
Bad	0	0	5	10
Worst	10	25	15	10
No response	20	15	10	40

4.5 Response Data Clustering

To obtain a clearer picture of the responses, it is necessary to group the scaling system. The information is then presented in the form of bar charts as given below:

4.5.1 Ease of Learning

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

From Figure 1 below, we can see that a total of 95 respondents indicate VB to be either very easy, relatively easy or okay to learn as compared to 75 responses for each of C++ and Java. Again this is probably true because VB has the "drag and drop program generating" facility in which you simply drag an object such as a Combo box unto the desktop, and then VB translate to equivalent program lines.

Pascal however, tops in the "Don't know + No response" bar. This could be due to the fact that it is outdated by now.

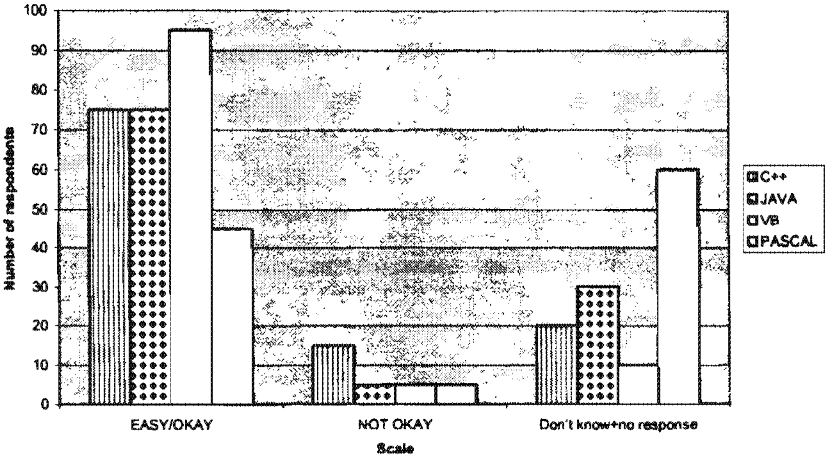


Figure 1: Ease of Learning

4.5.2 Ease of Use under Pressure

From Figure 2 below, more respondents (90) indicate VB to be easier to use under pressure, followed by Java and then C++. This is due to the "drag and drop facility" in VB as well as the windows development environment for both VB and Java which, in addition, also has Microsoft Disk Operating Systems (MSDOS) executing modes.

Pascal however runs mostly from the MSDOS prompt. To use it effectively, one has to be familiar with MSDOS commands and MSDOS development environment tools.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

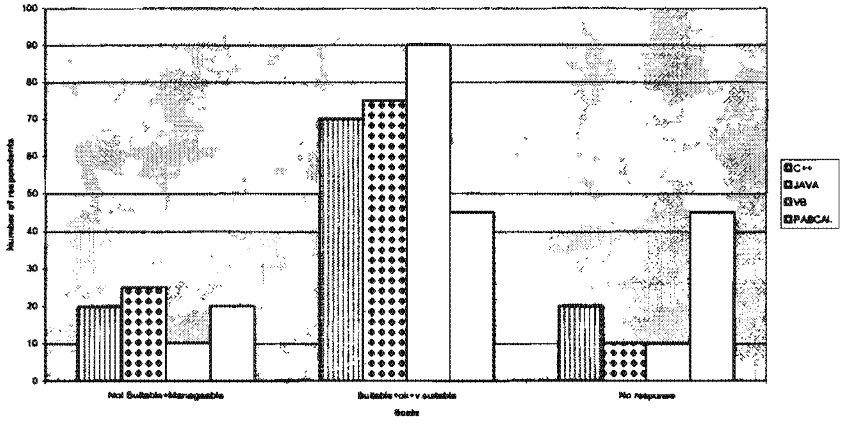


Figure 2: The ease of use under pressure

4.5.3 Suitability for Complex Jobs

From Figure 3 below, 90 respondents indicate VB to be best suitable for complex jobs, followed by Java and then C++. This is due to the fact that VB and Java can connect to other software tools like internet web pages and databases. VB does that via VB script and Java does that via javascript, servlets, and other middlewares.

Again, Pascal is less suitable for complex jobs, as those new tools were not yet available when Pascal came to being.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

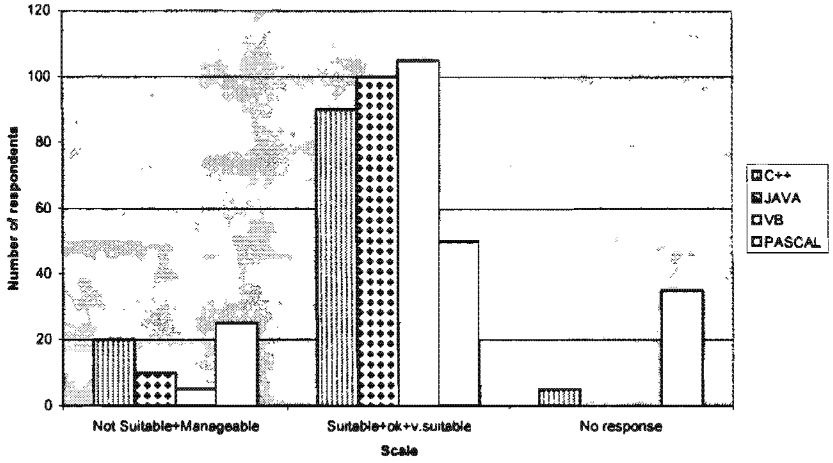


Figure 3: Suitability for complex jobs

4.5.4 Overall Rating for the Languages

From Figure 4 below, C++ and Visual Basic are considered to be equally best by 80 respondents. This is due to the fact that these two languages are the ones being currently used in most schools today. Java comes next with 70 responses, probably due to the fact that it is newly being introduced in most schools and in the industry.

With respect to the overall rating above, here again Pascal is rated the lowest as a programming language.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

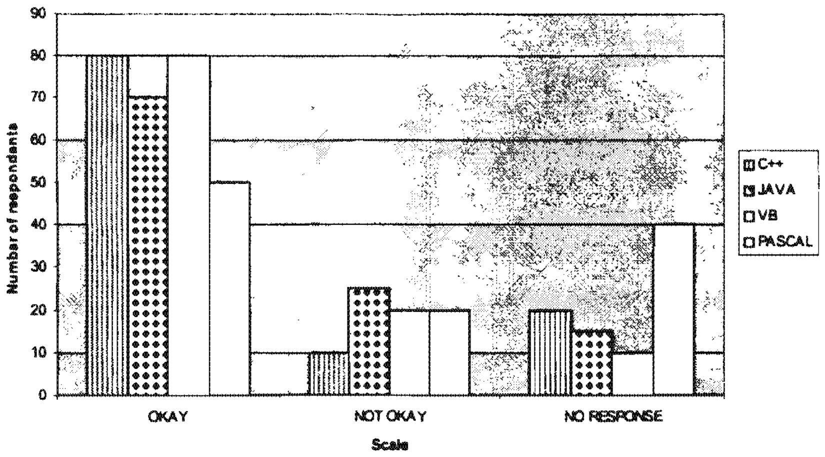


Figure 4: The overall rating for the programming languages

4.6 Line of Code (LOC) Estimation

From the completed questionnaires, the line of code was estimated for each programming language's solution to the given problem. The minimum line of code for each solution is given below:

Java: LOC = 10

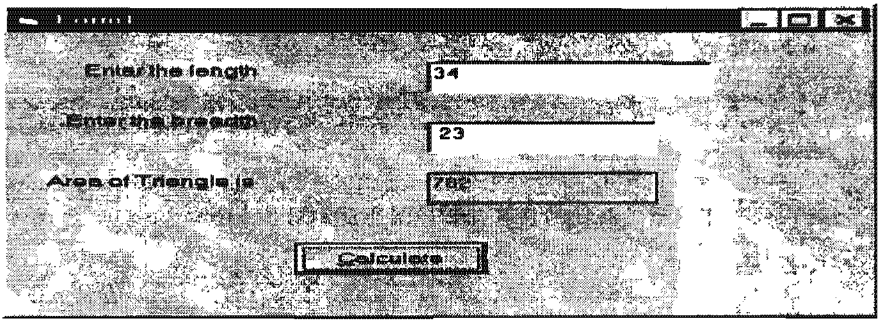
```
import java.lancs.*;
public class CalculateArea
{
public static void main (String[]args)throws Exception
{
int length, breath;
BasicIo.prompt("PLEASE ENTER THE LENGTH OF THE AREA");
length=BasicIo.readInteger();
BasicIo.prompt("PLEASE ENTER THE BREATH OF THE AREA");
breath=BasicIo.readInteger();
System.out.println("THE AREA OF RECTANGLE =" + length * breath + "|");
}//end of method main
}//end of class CalculateArea
```

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

VB: LOC = 8

```
Private Sub cmdCalculate_Click()
Dim intLength As Integer, intbreadth As Integer
Dim lngArea As Long
intLength = Val(txtLength.Text)
intbreadth = Val(txtBreadth.Text)
lngArea = intLength * intbreadth
txtarea.Text = lngArea
End Sub
```

This however is in addition to the "drag and drop facility" being used to design the input and output form as illustrated in the figure 5 below:



Enter the length 34

Enter the breadth 23

Area of Triangle is 782

Figure 5: The effect of the drag and drop facility

C++: LOC = 9

```
#include <iostream.h>
main ()
{
        int length, breadth;
        cout << "Enter length of triangle";
        cin >> length;
        cout << "Enter breadth of triangle";
        cin >> breadth;
```

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

```
        cout << "The area of the triangle is "<< length * breadth<<endl;
        return 0;
}
```

Pascal: LOC = 10

```
Program calcArea (input, output)
var
   length, breadth, area  : integer;
begin
   writeln ( 'Enter length of triangle')
   readln ( length );
  writeln ( 'Enter breadth of triangle')
  readln ( breadth );
  write ( 'The area of the triangle is  ',  length * breadth);
end
```

VB has the smallest LOC. In addition to the "drag and drop facility", this makes it easier to learn and use under pressure. C++ and Pascal also have reasonable LOC (8) minus the "begin" and "end" statement, making them easy to learn. Java however, has the longest LOC, making it more difficult for beginners to learn. However, this is compensated for by the flexibility it offers for complex jobs.

5. Conclusion

It was found that Pascal is simple to write for beginners, but not suitable for complex tasks.

VB is easy to use under work pressure possibly due to the "drag and drop program generating" facility. Additionally, VB has the smallest LOC. In addition to the "drag and drop facility", this makes it easier to learn and use under pressure. C++ and Pascal also have reasonable LOC (8) minus the "begin" and "end" statements, making them easy to learn. Java however has the longest LOC, making it more difficult for beginners to learn. Like VB, Java is suitable for complex jobs

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

and is considered very flexible as it interfaces with other web technologies like Java Servlets, and other middleware platforms.

The above findings therefore show that no single language can adequately satisfy all the requirements. But a careful combination of the languages can achieve the desired result. The study therefore concluded that the low pass rate for C++ is not due to C++ being an exceptionally difficult language, as there is no significant difference in the factors studied for C++, Visual Basic and Java.

6. Recommendations

From the above findings and conclusion, C++, Java and VB are all recommendable. VB is recommended for its ease of use under pressure due to the "drag and drop program generating" facility. C++ is recommended because of the fact that for simple programs, C++ has few lines of code (LOC), which would make it easer to learn. Java is recommended for its flexibility and suitability for complex jobs.

In my own opinion and experience, since C++ and Java have similar constructs, it will be alright for students to start with C++ in the first year and as soon as it starts becoming difficult at the second year level, they should move to Java which can do all C++ can do and more with a "gentle" language constructs. Visual Basic can be taught parallel to either C++ or Java in either first or second year respectively. Thus, a systematic combination of the programming languages could achieve the desired result of enhanced comprehension and potential capacity for future complex challenges. Further studies and analyses are therefore needed to pinpoint the cause of the low pass rate in C++ in previous years.

Further studies and analyses were initiated to pinpoint the cause of the low pass rate in C++ in previous years. This need is communicated to the Instructional Offering Committee (IOC) for programming subjects, which deliberated extensively and came up with the followings:

Program Design should be immediately translated into programming language in the same subject and not in a separate subject. This

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

will therefore mean that C++ programming in the first year will be divided into two components namely:

- Program design plus the lexical structure, and control structures.
- Functions, Arrays and Pointers.

By so doing, program design will serve as a background to the major programming concepts. Yet, the implementation of design into programming should not be delayed, thereby avoiding situations where students think program design is separate from programming. This, we hope will improve the comprehension of the programming concepts, thus leading to improved pass rate.

The same principles were found to be applicable to second year subjects. Object-oriented Analysis and Design should precede object–oriented programming as one need to analyze a problem before solving the problem. Therefore, Object-oriented analysis and Design should be incorporated into the beginning part of both "Java object-oriented programming" and "C++ object-oriented programming". The program design should be immediately translated into the applicable programming language in the same subject and not in a separate subject.

Similar principles are applicable to Visual Basic instructional offering. By the end of the second year, the students are thus expected to be versatile in the C++, Visual Basic and Java. This is in line with the current trends in the market where web-based development involves combining database with VBScript and JavaScript in server-based programming environment.

7. References

1. Bergin, J 1996. *Object Technology in the Classroom - Java as a Better C++* ACM SIGPLAN Curricular Patterns, pp. 21-27. USA : Wiley.
2. Corbetta, P 2003. *Social Research: Theory, Methods and Techniques.* U.K: Sage Publications.

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).

3. Dehinbo, J 2004. The Impact of Web-Based Middlewares on Training and Assessment through In-House Developed On-Line Testing System. *Proceedings of the Informing Science + Information Technology Education (I'SITE) Conference*, Australia. http://proceedings.informingscience.org/InSITE2004/027dehin.pdf .
4. Deitel, HM, PJ Deitel, & TR Neito 2001. *e-Business & e-Commerce: How to Program.* USA: Prentice Hall.
5. Hamilton, MA 1996. *Java and the Shift to Net-Centric Computing Practices* IEEE, pp. 31-39. USA.
6. Lerdorf, R. & K Tatroe 2002. *Programming PHP: Creating Dynamic Web Pages.* USA: O'Reilly & Associates Inc.
7. Sebesta, RW 1996. *Concepts of Programming Languages* 3rd Edition. USA: Addison-Wesley Publishing Company.
8. Zak, D 2001. *An Introduction to Programming with C++.* 2nd edition. USA: Course Technology – Thompson Learning.

Author's Contact Details

JO Dehinbo (DehinboOJ@tut.ac.za JDehinbo@yahoo.com)
Computer Studies Department
Tshwane University of Technology, Pretoria, South Africa

Reproduced by Sabinet Gateway under licence granted by the Publisher (dated 2010).