



A Critical Evaluation of Java as a Good Choice for Introductory Course

*Osisikankwu P. U.; *Ikedilo O.E.; **Madubuike C. E;

*Ebonyi State University, Abakaliki, Ebonyi State, Nigeria

**Akanu Ibaiaam Federal Polytechnic, Unwana, Ebonyi State, Nigeria

ABSTRACT

The choice of programming language for introductory course is very important for future development of Information Technology (IT) professionals. Even though a large number of programming languages have been developed over the years, there is still not yet a consensus for suitable programming language for introductory course. Thus, choosing appropriate programming language for introductory course is still demanding. This paper critically analyzes the desirable features of Java programming language making it suitable as a choice for first programming course. The analysis was carried out through a detailed research and evaluation of information from secondary sources. The internet search engine was adequately employed to access journal papers from both national and international sources. The study enables computer educators and students to understand the trendy features of Java programming language, its popularity in industry and its suitability for introductory course based on selected criteria. It also keeps academicians and researchers abreast of the desirable attributes of Java programming language.

Keywords: Java; programming education; OOP; programming language ranking; language popularity; features of java

1.0 INTRODUCTION

As it stands, almost every aspect of the modern society relies on Information Technology (IT). Computing has happened to be the central technology of our time, making computer-programming education very vital. Computer Science is extremely vivacious field in this current time. Existing technologies continually becomes out-of-date as soon as new technologies are introduced. According to (ACM, 2001), the speedy growth of the discipline has an intense effect on computer science education. It is clear that the choice for suitable programming language for introductory course is important in development of an IT professional (Abelson et al, 1998). Several programming languages belonging to different paradigms have been introduced over the years. However, not many languages survive for more than ten years (Vujosevic-Janjic & Tosic, 2008). There is still not a consensus for most suitable programming paradigm and programming language for introductory courses, after long years of teaching programming. Programming languages like C++, C#, Java and Visual Basic are the most popular and widely taught in programming education nowadays (TIOBE, 2015). It is very important for programming education to consider the languages that are most widely used in the industries for teaching programming. Thus, creating a consensus for introductory programming course in computer education will help in faster development of the

language and reduce the difficulty of changing from one language to another.

The objective of this paper is to critically discuss why Java programming language is most suitable for introductory programming course according to the criterion identified by (DeRaadt, Watson & Toleman, 2003) for choosing introductory programming course. The study will also enable computer educators and students to understand the trendy features and popularity of java programming language.

2.0 ACADEMIC REVIEW

There have been several debates in the academic society on what paradigms and languages to be used as first choice for introductory programming course for education (Kellehar & Pausch, 2005; Bruce, 2005). According to Kolling (1999), Farooq, et al (2014) and Muhammed (2012), several researchers have suggested some criteria and methods for assessing programming languages as a suitable first programming language for education; however, Vujosevic-Janjic and Tosic (2008), overviewed some of the most relevant issues in introductory course for programming education. They also suggested that features like establishment and maintenance costs, compiler availability, language popularity, textbook quality are important factors in choosing first programming language for computer education. Farooq, et al, (2014) highlights different pedagogical approaches suitable for introductory language course, and gave advantages and

disadvantages of those approaches. Thus, many other studies like focused on comparative analysis/investigation of the suitability of existing programming languages as first choice programming language (Peers, 2007; Farooq, 2014; Muhammad 2012). More so, as Irimia (2001), in his paper compared C++ and Java based on most suitable language for first programming course.

This paper also investigates, through literatures and internet sources, the most popular programming language in the industry. The study reveals that the most required programming language in the industry according to the census conducted by (DeRaadt, Watson & Toleman, 2003) as of 2002 to 2003 was java programming language. Again, a recent online survey by TIOBE Programming Language Index (2015) also reveals Java had the most increased popularity than any other programming language in 2015.

TIOBE index is updated once a month, and indicates the most popular programming language. It is rated based on number of third party vendors, courses and skilled engineers worldwide. The ratings are calculated using popular search engines like Google, Yahoo, Bing, YouTube, Wikipedia, Amazon and Baidu. It was said to be assistive tool for making strategic decision about what programming language to use when building new system software; and as well, for checking programming skill updates. TIOBE Index (2015) suggests Androids as the major reason for Java's popularity, there has been an

increased demand for Java Android programming since October 2014.

In addition, the survey of programming courses in the USA by Davies et al (2011) reveals that Java has continued to hold the place of the most popular language and that it is the reason for java popularity in industries.

2.1. HISTORY OF JAVA

Java is a powerful programming language developed by James Gosling and a group of Engineers at Sun Microsystems of USA in 1991 (Arnold et al, 2006), with the initial aim of developing a language for consumer devices like TV, toasters and other home appliances. They also wanted a language that will be small and can be able to run on different platform. The original name of the language was Oak, and was later renamed to Java. Java is an object oriented programming language; a high level third generation programming language just like C, Perl and others. Compared to other languages; Java lets programmers write special programs that can be played on web browser and download from the internet using applets.

Java gained public attention through Applets; and can be used to develop web applications, mobile applications, embedded applications and desktop applications. Since the introduction of Java programming language in 1995, Java has incessantly been improved year after year with new features as seen in table 1 (Tasaka & Ginbayashi, 2004).

Table 1: Versions, Upgrades and Features of Major Java Release

Release Date	Version	Name	Features
May 1995	Oak to Java	Java	Compiled and Interpreted; Platform Independent and portable; Object-oriented; Robust and secure; Distributed; Familiar, simple and small; Multithreaded and Interactive; High performance; Dynamic and Extensible
Jan 1996	1.0	JDK 1.0	Object orientation; Cross-platform support; Network compatibility
Feb 1997	1.1	JDK 1.1	Internationalization; JavaBeans; Remote Method Invocation (RMI); A new event model; Java Database Connectivity (JDBC)
Dec 1998	1.2	Java 2 (Jdk 1.2)	Java Foundation Classes (JFC); Input Method Framework (IMF); Common Object Request Broker; Architecture (CORBA) support; EJB; Java Plug-ins
Jul 2000	1.3	Java 2 (J2SE 1.3, J2EE 1.2, J2ME)	HotSpot Client; Virtual Machine (VM); Java Naming and Directory Interface (JNDI); RMI/Internet Inter-ORB; Protocol (IIOP); Security enhancements
Feb 2002	1.4	Java 2 (J2SE 1.4, J2EE 1.3, J2ME)	XML support; Java Web Start; A new I/O; A logging API; Assertion; Java Secure Socket Extension (JSSE)
Sept 2004	1.5	Tiger, J2SE 5.0	Generics; Metadata; Autoboxing/Unboxing; Enumeration; Varargs; Enhanced for Loop (for each); Static Import
Dec 2006	1.6	Mustang, Java SE 6	Support for older Win9x versions; Script Language Support (JSR 223); Improve performance for the core platform and swing; Improve Web Service support via JAX-WS (JSR 224); JDBC 4.0 support (JSR 221); Java Compiler API (JSR 199); Upgrade of JAXB to version 2.0; Support for pluggable annotation (JSR 269); GUIs improvement; JVM improvements; And later updates to Java SE 6
July 2011	1.7	Dolphin, Java SE 7	JVM support for dynamic languages; Compressed 64-bit pointers; Currency utilities under JSR 166; New file I/O library; Timsort instead of merge sort for sorting collections and arrays of objects; Java Deployment Rule sets; Upstream updates to XML and Unicode; New platform APIs for the graphics features; Library-level support for elliptic curve cryptography algorithm; An XRender pipeline for Java 2D; Later updates of Java SE 7 by Oracle



Mar 2014	1.8	Java SE 8	JSR 223, JEP 174: Project Nashorn; JSR 335, JEP 126: Language level support for lambda expressions; Unsigned Integer Arithmetic; JSR 308, JEP 104: Annotation on Java Types; JSR 310, JEP 150: Date and Time API; JSR 337, JEP 120: Repeating annotations; JEP 178: Statically-linked JNI libraries; JEP 122: Remove the permanent generation; JEP 153: Launch JavaFXApplications; Later updates on Java SE 8
----------	-----	-----------	---

2.2 FEATURES OF JAVA PROGRAMMING LANGUAGE

There are many desirable features making Java a good choice for an introductory course (King, 1997).

I. Object Oriented Programming (OOP)

Java is highly OOP language that uses object as almost everything, data and programming code exist in classes and objects. Even to write a simple program in Java, it is fundamental to understand the basic principles in OOP. OOP took the best concept of structured programming and other several new concepts, and has its own way of organizing programs unlike the procedural and structured programming technique. Currently, OOP is most widely used programming paradigm. (Declue, T. 1996; Lattanzi & Henry, 1996; Bergin, 2000; Guzdial, 1995) identified two main reasons that convinced many instructors to believe that object-oriented programming should be taught as first programming courses. (1) The great importance and popularity of the paradigm. (2) The difficulty that arise from the paradigm shifting. Having more experiences in one paradigm makes it more difficult to shift to another paradigm. Some of the characteristics of OOP as in Java are class, inheritance, abstraction, polymorphism and encapsulation. As an OOP programming language, Java enjoys the benefit of enhanced software maintainability; faster development; reduced cost of development; enhanced software development productivity; and higher-quality software (The Saylor Foundation, 2012).

II. Platform Independency and Portability

Java programming language introduces the concept of platform independency and portability; thus, enabling applications to be written once and run on different hardware and Operating Systems (OS) such as Windows, Linux, and Macintosh without any modification. Java language provides platform independence unlike other programming language whose code needs to be rewrite or recompiled for every single platform. Java uses its bytecodes and Java Virtual Machine (JVM) to achieve platform independence. This is because, with bytecodes, Java instructions are run by the JVM at the hardware abstraction layer instead of the real hardware. Then JVM convert the

bytecodes into machine instructions for the intended hardware. Such that upgrades and/or changes in operating systems and/or system resource will not affect java application. For instance, an upgrade from Windows-Vista OS to Windows7 OS will not affect java application unlike C++ language where an upgrade in the OS libraries will force an upgrade in the C++ application. Java follows the "Write once, run anywhere" approach. It is this feature that makes Java a trendy language for programming on Internet (Hubbard, 2004).

III. Distributed and Secured

Java is a language developed to run not only as a stand-alone application but also for developing a network/distributed applications. Java contain extensive library of routines that can easily deal with TCP/IP protocol like http and ftp. This makes network connections much easier than with C/C++. There are no much differences between distributed applications and non-distributed applications in Java - they look very much alike. Both cases used basically related programming model. Just that distributed case requires that applications consider the effect of network failures. Java application can read and write objects over the internet through URLs just as easy as accessing data on a local file system.

Thus, emphasis has been placed on network systems security. Java indeed uses authentication technique based on public key encryption to address the issue of security through enabling of construction of virus free and corrupt free systems [1]. Java uses the concept of exception handling that restrains serious errors and reduces all sorts of threat connected with crashing systems. The absence of pointer in Java also ensures that programs cannot get right of entry to memory location without proper approval. More so as seen in the frequent updates in table 1, Java includes enhanced security features from version 1.3.

IV. Compiled and Interpreted

Unlike most programming languages, Java uses compiled and interpreted approach to execute its program; making Java programs a two-stage system. Java codes are first compiled into bytecodes by java compiler called javac; these bytecodes which are generic in nature like their



source code counterparts are later interpreted by individual client machines using their program called Java virtual Machine (Hubbard, 2004). To compile and run a program, one needs a Java Software Development Kit (SDK) of some kind. Java Runtime Environment (JRE) enables the run of existing Java programs, but not to write a program.

V. **Multithreaded and Interactive**

Multithreading is a way of building applications with multiple threads. Java maintains multithreaded program, such that it manages multiple tasks simultaneously (Horstmann & Connell 1999). That means we need not wait for the application to complete one task before starting next task. The feature is very suitable for graphic applications. Java has a sophisticated set of synchronization primitives that are based on the widely used monitor and condition variable paradigm that was introduced by C.A.R. Hoare. The integration of this concept into the language makes it more easy to use and robust. Java Runtime Environment (JRE) creates a real-time behavior and interactive responsiveness, which is one of the benefits.

Using its multithreaded techniques, Java codes are segments into smaller parts called threads to be executed by java compiler simultaneously in sequence and timing manner. These Java codes could support both Command Line Interface (CLI) and Graphical user Interface (GUI).

3.0 WHY JAVA AS INTRODUCTORY COURSE

Students are more enthusiastic when studying a language they feel will make them more employable and/or marketable. It is clear that the choice of language as introductory course is a key decision in this modern world. For that reason, this study discusses in favor of Java programming language as best choice for introductory programming language. Our discussion is based on the six criteria given by the Australian Universities on the reason for their choice of introductory programming language (De Raadt et al, 2003).

i. Industry/Marketable Use: The boost in the application area of Java has contributed to the popularity and industrial use of the language. Java has expanded to include enterprise mission critical systems and mobile appliances. The integration of Java in most major operating system platforms has practically placed Java above other language. Java can be used to develop application that can be built into consumer electronic devices

such as PDAs, Televisions, cards, boxes and cell phones. Almost more than 95% of industry uses Java card technology. Therefore student who learn Java as first programming language will not only have the wherewithal to develop varieties of applications but also have the opportunity of been employed as IT professional in over 95% of industry using java technology. Java has been proven strong enough for practical day-to-day use and is certainly better than any of the alternatives.

ii. Pedagogical Benefits of Language: Although it can be argued that programming language in introductory course is a means of presenting concepts and not final results. According to (Tucker, 1996), decisions concerning the suitability of a language for an introductory course in programming should not only consider the requirements of industry but also should take into account the commitment to sound pedagogical practice. In fact, Java became the programming language used to demonstrate all programming concepts and programming issues in most universities. Java has a lot of programming concepts to clarify pedagogical benefits alongside its rich library environment that explores almost all area of computer science ranging from Networking, security, GUI, internationalization, threading, reflection etc used in modern-day software. Java represents advanced step of computer technology where programming principles can be grouped in much easier approach. Therefore, Java as introductory course will expose students to virtually major areas of computer science.

iii. Object Oriented Language: Java is a general-purpose object-oriented programming (OOP) language that supports the concept of object-oriented programming. According to (Declue, 1996; Lattanzi & Henry, 1996; Bergin, 2000; Guzdial, 1995), object-oriented programming is most suitable programming paradigm to be used as introductory course because of its popularity and great importance. As an OOP language, Java supports reusability of written code, enhanced software maintainability, fast development, reduced cost of development, high quality software, and enhanced software development productivity. Therefore, Java as first programming language places student on the latest and most advanced programming paradigm.



- iv. **Graphic User Interface (GUI) and Multimedia Capabilities:** Java is a very powerful language that enables easy integration of multimedia programs into applications such as images, sounds, graphics and videos. As the world advances in technology, many multimedia devices are developed, like 3D accelerator cards, Bluray-Rom drives, sound cards and other new graphic technology capabilities. Java language enables students to create a multimedia program with easy through extensive multimedia facilities that enable a programmer to start developing powerful multimedia application immediately. Java supports a lot of image, audio and video formats. Java IDEs provides user interface-based support for application development. Java IDEs uses SDK to provide more futures like Java-aware editor, graphical shortcuts to standard SDK tools, support for visual debugging and others. It is clear that students seem to appreciate and enjoy java modular approach and writing GUI-based code, which can be fun and enjoyable, and a valuable real-world skill in itself.
- v. **Availability/Cost to Students:** Another advantage of java is the availability and free cost of the tools needed to develop and test java programs. Sun Microsystems makes Java Development Kit (JDK) available for free download over the Internet at www.javasoft.com. Students and educators can download it for free. JDK includes the Java compiler, interpreter and other simple tools students can adequately use for programming assignments. However, there are other pleasant program development environments that can still be purchased at a cheaper price for learning. Also, unlike C++ and some other programming languages that are proprietary, Java is open source programming languages with different individuals and groups contributing to its development. OpenJDK is an open-source implementation of Java platform, standard edition (Java SE) specifications. Sun released complete JDK version in May 2007 for Java SE to OpenJDK Community as free software under the GNU General Public License version 2 (GPLv2).
- vi. **Easy to Find Appropriate Texts:** Everything about Java is open, from the language to the standards, virtual machine, core libraries and development tools.

Students can easily access and download appropriate texts on Java language over the Internet without restriction. Thousands of java tutorials can easily be found and copied from Internet that teaches how to program in Java both for beginners and advanced. There are lots of free Java source codes which programmers or students can easily download and modify (code reuse) to suite their programming needs. Java has in J2SE 1.5 extends the programming language specification which carefully realize a high degree of ease of development and improve ease of learning and ease of mastering. With the new technologies, Java will be adapted by wide range of programmers.

- vii. **OS/Machine Limitations of Dept/Platform Independent:** As described earlier, Java programs are compiled to a portable intermediate form known as bytecodes, rather than to native machine language instructions. The Java Virtual Machine runs a Java program by interpreting these portable bytecodes. In-fact, platform independence is one of the compelling benefits of Java. Java runs on major hardware and software platforms such as Macintosh, Windows, and Unix. Unlike other software, Java software can run on any platform without been modified. For instance, Java 2 SDK is a highly structured and secured development environment that has been executed and ported on almost all the main OS platforms. Also, Java does not occupy memory space like other languages that is specifically build to run on a particular platform.

Further study conducted in 2010 by Mason et al (2012), revealed the emergence of many new criteria for choosing first programming language to be taught. This include the availability of a community and online help, platform independence, extendibility and libraries available, ease of installation and interpreted language, with no need to compile. These new criteria reflect the rise of the open source community over the time period, as well as perhaps a greater choice of operating systems by students. For this reasons one will conclude that Java is a suitable choice of language for introductory course. Java is open source with huge community and online help; platform independent; easy to install; extendable and available libraries; and interpreted language.

4.0 CONCLUSION

Java programming language is a powerful object oriented programming language with great admirable



features and very largely accepted and used in industries today, both in developing web applications, embedded applications, mobile applications and desktop applications. But, has still not been accepted by many computer programming education as a first choice for teaching introductory programming course. This study has critically analyzed the features of Java programming language and has discussed based on the selected criteria given by the Australian Universities and Mason et al (2012) for choosing introductory programming language. In every ramification, Java stands fit as a suitable and benefiting language for introductory course. We suggest that industrial benefit/popularity of a language cannot be completely excluded as a determinant criterion; because students are more enthusiastic when studying a language they feel will make them more employable/marketable. Hence java is a good first choice for introductory programming language for now and the future.

REFERENCES

- [1] Abelson, H., Dybvig, R.K., Haynes, C.T., Rozas, G.J., Adams N.I., Friedman, D.P., Kohl-Becker, E., Steele G.L. Jr., Bartley, D.H., Halstead, R., Oxley, D., Sussman, G.J., Brooks, G., Han-Son C., Pitman, K.M., and Wand, M., (1998): Revised Report On The Algorithmic Language Scheme, Higher- Order And Symbolic Computation, vol. 11, pp.7-105.
- [2] Arnold, K., Gosling, J., and Holmes, D. (2006): The Java Programming Language, Addison-Wesley
- [3] Bell, D. (2002): Visual Basic.Net as a First Language: An Evaluation. ACM Sigcse Bulletin, vol. 34(4); pp.107-108.
- [4] Bergin, J. (2000): Why Procedural is the Wrong First Paradigm if OOP is the Goal.
- [5] Biancuzzi, Federico and Warden, Shane (2009): Masterminds of Programming. O'reilly Media, Inc. pp.242-246. ISBN0-596- 51517-0.
- [6] Bruce, K. B. (2005): Controversy On How To Teach Cs 1: A Discussion on the Sigcse-Members Mailing List. Inroads - The Sigcse Bulletin 37: pp.111-117.
- [7] Davies, S., Polack-Wahl, J.A. & Anewalt, K., (2011): A Snapshot of Current Practices in Teaching the Introductory Programming Sequence. Sigcse 11 Proceedings of the 42nd ACM Technical Symposium on Computer Science Education. ACM Press, pp. 625 - 630.
- [8] De Raadt, M., Watson, R. and Toleman, M. (2002): Language trends in introductory programming courses. Proc. Informing Science and IT Education Conference, Cork, Ireland, Cohen, E. and Boyd, E. (Eds). InformingScience.org
- [9] De Raadt, M., Watson, R. and Toleman, M. (2003): Language Tug-Of-War: Industry Demand and Academic Choice. Proc. Fifth Australasian Computing Education Conference (Ace2003), Adelaide, Australia, 20:137-142, Greening, T. And Lister, R. (Eds). Conferences in Research and Practice in Information Technology.
- [10] De Raadt, M., Watson, R. and Toleman, M. (2004): Introductory Programming: What's Happening Today and Will There Be Any Students To Teach Tomorrow? Proceedings of the Sixth Conference on Australasian Computing Education. Dunedin, New Zealand, 3, Australian Computing Society, Inc.
- [11] Declue, T. (1996): Object-Orientation and Principles of Learning Theory: A New Look at Problems and Benefits, Proc. 27th Sigcse Technical Symposium on Computer Science Education, Pp.232-235-86.
- [12] Farooq, Muhammad Shoaib, Sher Afzal Khan, Farooq Ahmad, Saeed Islam, and Adnan Abid.(2014): "An Evaluation Framework And Comparative Analysis Of The Widely Used First Programming Languages". Plos One 9(2); E88941.
- [13] Gauravjindal, Poojakhurana, Tanugoel (2013): Comparative Study of C, Objective C, C++ Programming Language International Journal of Engineering and Computer Science ISSN: 2319-7242 vol. 2 (1) pp. 202-206
- [14] Gupta, D., (2004): What is a Good First Programming Language? Crossroads, vol.10(4), pp.7.
- [15] Guzdial, M. (1995): Centralized Mindset: A Student Problem with Object-Oriented Programming, Proc. Sigcse95.
- [16] Hubbard, J. R. (2004). Schaum's Outline of Theory and Problems of Programming with Java, The McGraw-Hill Companies, USA
- [17] Holtz, N. M., and Rasdorf, W. J. (1998): An Evaluation of Programming Languages and Language



Features for Engineering Software Development, Engineering with Computers; vol. 3(4); pp.183–199.

[18] Horstmann, Cay S. and Connell, Gary. (1999): "Core Java". "Fundamentals". Sun Microsystems Press. A Prentice Hall Title. Palo Alto, California. vol(1)

[19] Irimia Andrei (2001): Enhancing the Introductory Computer Science Curriculum: C++ or Java? Student Paper Lipscomb University Nashville, Tennessee

[20] Kelleher, C. and Pausch, R. (2005): Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. ACM Computing Surveys (Csur), 37(2):83-137. ACM Press.

[21] Kelleher, C., Pausch, R. Lowering (2005): The Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. ACM Computing Surveys, vol. 37(2), pp.83-137.

[22] King, K. N. (1997): The Case for Java as a First Language, Proc. 35th Annual ACM Southeast Conference, pp. 124–131.

[23] Kölling, M. (1999): The Problem of Teaching Object Oriented Programming Part 2: Environments, Journal of Object Oriented Programming, vol. 11(9), pp.6-12.

[24] Lattanzi M. R. and Henry, S. M. (1996): Teaching the Object-Oriented Paradigm And Software Reuse: Notes From An Empirical Study, Computer Science Education, vol. 7(1), pp.99–108

[25] Mason Raina, Cooper Graham, DeRaadt Michael, (2012): Trends in Introductory Programming

Courses in Australian Universities – Languages, Environments and Pedagogy; Proceedings of the Fourteenth Australasian Computing Education Conference (ACE2012), Melbourne, Australia

[26] Muhammad Shoaib Farooq, Sher Afzal Khan, Adnan Abid (2012): “A Framework for the Assessment of a First Programming Language”, Journal of Basic and Applied Scientific Research; vol. 2(8); 8144-8149.

[27] Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., and Paterson, J. (2007): A Survey of Literature on the Teaching of Introductory Programming, ACM Sigcse Bulletin; vol. 39(4), pp. 204-223.

[28] Tasaka, S., and Ginbayashi Jun (2004): Latest Progress and Trends in Java/Ejb Technologies Fujitsu Sci. Tech. J., vol. 40(1) pp.85-93

[29] The Saylor Foundation (2012). Advantages and Disadvantages of Object-Oriented Programming, <http://saylor.com/courses/csc101/#2.1.2> accessed on 03/09/2015

[30] TIOBE Software (2015): Tiobe Programming Community Index - Long Term Trends. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> Accessed 20 April 15

[31] Tucker, Allen B. (1996): "Strategic Directions in Computer Science Education". ACM Computing Surveys, V28, N4 P836 (10).

[32] Vujosevic-Janjic Milena and Tomic DuSan (2008): Role Of Programming Paradigms In The First Programming Courses The Teaching Of Mathematics, Vol. Xi, 2, pp. 63-83