A QUALITATIVE STUDY OF MAJOR PROGRAMMING LANGUAGES: TEACHING PROGRAMMING LANGUAGES TO COMPUTER SCIENCE STUDENTS

Ghazala Shafi Sheikh[1]
Department of Computer Science, Indus University
Noman Islam[2]
Department of Computer Science, Iqra University, Karachi

ABSTRACT

Programming is a fundamental course that is taught to every computer science during their initial semesters. The course introduces the students to basics operations and architecture of computers, and also polishes the problem solving skills of students. Besides these advantages, programming language serves as a fundamental tool for analyzing, studying and understanding advanced concepts of computer science that the students are taught in later semesters of their undergraduate studies. Therefore, the selection of a programming language for teaching to computer science students is extremely important. During the past few years, there have been numerous programming languages evolved such as COBOL, FORTRAN, Algol, Miranda, Oberon, Ada and Java etc. With the passage of time, some these languages have lost prominence while several new languages have emerged. Therefore, the selection of a programming language for teaching has always remained an important research question for academicians. In this paper, a comparative analysis of contemporary programming languages is performed. After a careful examination of current curriculum and market demands, we have selected C/C++, C#, Java, Pascal, GW Basic and JavaScript for comparison. The objective of this study is to determine which programming language languages should be taught to computer science students at introductory level. The paper analyzes the selected programming languages based on different parameters and provides recommendations on the selection of programming language.

Key Words: Programming Language, Teaching, Education, Computer Science, Comparative Study, Introductory Programming Language.

[1] Ghazala Shafi Sheikh   : ghazala.shafi@indus.edu.pk
[2] Noman Islam   : noman.islam@iuk.edu.pk

1. INTRODUCTION

Which programming languages should be taught to computer science students? This question is raised often during the development of curriculum for computer science and software engineering students. It has been found that programming languages have the highest drop rate (Robin's et.al, 2003). According to various studies, students have problems in comprehending what is happening in memory, instruction cycle, and overall picture of the execution of program (MILNE, I. and G. ROWE, 2002). As computer science is a rapidly evolving field, the answer to this question varies with the passage of time. In this paper, an updated analysis of major programming languages of recent time has been performed. The selection of the programming language is based on surveys such as of various introductory level programming courses taught at college/ undergraduate level (Davies, et.al, 2012). This paper compares C/C++, C#; Java, Pascal, GW Basic and JavaScript based on different criteria and provide recommendations (Schulte, C. and J. Bennedsen, 2006).

2. LITERATURE REVIEW

In literature, several studies have been reported that discussed merits and demerits of various programming languages and the issues involved in selection of a programming language for teaching. Schulte and Bennedsen gathered teacher's opinion about what topics should be taught in programming courses (Schulte, C. and J. Bennedsen, 2006). Milne and Rowe analyzed what are the topics in introductory level courses difficult to be comprehended by undergraduate students. Davies et al. provided a nationwide survey of various languages and techniques being taught in introductory level programming courses. According to the authors, Java is the most widely used language for teaching programming. Mason et al. (Mason et.al, 2012) analyzed different introductory level programming courses in Australian Universities to determine the trends in programming language, Integrated Development Environment (IDE), paradigm and topics coverage. Robins et al. provided a review on programming languages and identify topics related to teaching novice users (Sebesta, et.al, 2014).

In various studies, different programming languages have been analyzed based on their features for novice programmers. A brief discussion on various programming languages of choices for beginners has been provided by (Lisa Eadicicco, 2014). An overview of various programming languages of choices for modern days has been provided in (Rebecca Hiscott, 2014). In some of the studies, specialized tools have been developed for teaching programming to computer science students in more effective manner. In a tool called SAUCE has been developed for students and educators to learn parallel programming (Moritz Schlarb, et.al, 2015). A list of various tools to teaching programming to kids has been provided in. According to, the prevalence of mobile gadgets demands that programming should be taught directly on mobile devices. A tool called Touch Develop has been developed realizing this need (Nikolai Tillmann et.al, 2012).

It has been concluded from literature review that research has not analyzed currently popular programming languages for their aptness for introductory programming teaching.

languages based primarily on their features as well as the support they provided to be used in teaching various concepts of computer science (Rebecca Hiscott, 2014).

2.1   Criteria for Comparison

The criteria for comparison are based on the response of various computer science teachers to adopt a particular language for teaching. We mainly adopt the criteria provided in [6] and [5]. Following criteria have been considered for evaluation of language:

Simplicity: In order to be taught as introductory level course, a language should be very simple and closer to natural language. A language closer to human language is readable and easier to be understood by a novice user.

Writability: A language should provide a comprehensive set of constructs and APIs to be useful for general purpose as well as specific programming tasks.

Reliability: A language should have good reliability. In particular support for pointers, union and aliasing etc. should be discouraged. Additionally, support for assertions, error checking and exception handling should be provided by the language such that abnormal conditions of the program are properly taken care of by the programmers.

Appropriate Data Structures: A language should have support for variety of primitive data types and also have the provision to construct user defined data types as needed.

Availability/ Cost to students: The cost of development platform should be low. Ideally, development tools, compilers, interpreters and Integrated Development Environment (IDEs) should be freely available as open source tools.

Market demand: It is recommended that the language being taught to computer science student should have high demand in market. There should be appropriate jobs available for developers.

Community Support: The language should have documentation, tutorials, and community's support and developer forums widely available.

OS/Machine Limitations: The language should have minimum platform requirements. In particular, it should be easily able to run on common operating system.

Extensions/Libraries available: The language should have extensions available in abundance for specific tasks such as drivers for hardware interfacing, database connectivity APIs, parsers, GUI libraries etc.

Coverage: The language should have sufficient coverage to be useful to teach computer science concepts. These topics include: object oriented programming, multithreading, I/O, mobile computing, databases, system level programming etc.

3. A COMPARISON OF MAJOR PROGRAMMING LANGUAGES

This section compares the selected programming languages based on the criteria outlined above.

3.1   Simplicity

   A language should have a simple, easy to use syntax that is closer to natural language. Pascal and GW Basic have a very simple syntax. It uses self-explanatory keywords such as begin, end, to write blocks of code. JavaScript, C++, C# and Java has almost similar syntaxes that have now been widely accepted for general purpose programming language. Figure 1 compares sample Hello Word code for Pascal, GW Basic and Java.

```
Program Hello Word;
Begin
 Write ('Hello World.');
End.
```

a. Pascal

```
10 print "Hello World"
Run
```

b. GW Basic

```
public class Test {
public static void main(String args[]) {
    System.out.println("Hello World");
        }
        }
```

c. Java

Figure 1: Sample programs in Pascal, GW Basic and Java

3.2   Writability

The writability of a language stands for the availability of different types of constructs to easily write different types of programs. All of the languages provide basic constructs for looping, conditional statements, procedures etc. However, GW Basic, Pascal, C and JavaScript don't provide adequate support for data types. In C, for instance, it is very difficult to operate on strings. Similarly, support for Boolean data type is not available. User defined data types can be created in C language using struct, typ def, enumerations etc. In JavaScript, user defined data types can be created using objects. Java and C# provide good support for data types such as integer, long, float, double, character, Boolean and String data types.

3.3   Reliability

The availability of pointers in Pascal and C impacts its reliability as this can create dangling references. C# also allows the use of pointers but in unsafe region. An unsafe region provides the flexibility to the program to perform various restricted operations, put it puts additional burden on programmer to handle various low level issues (such as memory leakages, dangling pointers and type safety). C#, Java and JavaScript support exception handling. Figure 2 shows exception handling code for Java.

```
try {
    FileOutputStream fos = new FileOutputStream ("C:/data.txt");
  fos.write("Sample data");
}catch (IOException e) {
  System.out.println("IO Exception occurred");
}
```

Figure 2: Exception handling in Java

```
try {
        myFun();
}
catch (e) {
        document.writeln("Exception occurred");
}
```

Figure 3: Exception handling in JavaScript

However, JavaScript is not type safe language. The same can be said for language. C language doesn't provide support for exception handling; however, assertions can be used. In addition, the global variable errno can be used to track the error. Figure 4 shows how error no can be used in C language. The use of go to in GW Basic also adds to unreliability in program.

```
#include "errno.h"
#include "string.h"

int o = read(ffd, buffer, 1);
if(o == -1) {
    printf("Error! %s\n", strerror(errno));
}
```

Figure 4: Using errno to handle errors in C language

3.4   Data Structures

Java provides eight basic data types i.e. byte, short, int, long, char, float, double and Boolean. In addition, there is support for String and long range integers and real numbers in the form of available classes. User defined data types can be created using classes. C language has support for int, long, float, char and double, but doesn't provide Boolean data type. String variables are manipulated using character array or character pointer where the end of the string is delimited by null character ('\0'). User defined data types can be created using structure, type def, enumerations etc. C# provides support for various data types such as byte, short, and integer, long, single and double precision real numbers, decimal, Boolean, date, char and string. User defined data types can be created using structure, enumeration and classes. Pascal has support for string, integer, and real, Boolean and character data types. User defined data types in Pascal can be created using enumerations, sub-ranges, records and set. GW Basic provides support for string, integer, and single precision and double precision real numbers. In JavaScript, variables are not explicitly declared to be of any type, but their types are determined based on the value they hold. A variable can hold numeric, string or Boolean value.

3.5   Availability/ Cost to Students

Java Development Kit is available free of cost. Different integrated development environment such as Eclipse and Net Beans are available free of cost. However, some development toolkits such as J-Builder are available with licensing fee. The IDE and compiler for C, Pascal and GW Basic are available free of cost. The most widely used development platform for C# is Visual Studio which is available free of cost for developers. The .Net framework required to run C# program is also available free of cost. JavaScript can be easily run on any standard browser. The most widely environment used for development for JavaScript is Web Storm which is available with some licensing fee.

3.6   Market Demand

According to a survey by Business Insider, Java is amongst the highly paid jobs in the market. JavaScript developer gets less salary than Java but more than C while C# is at the bottom of the developer's salary. According to another survey, Java has the

maximum job market followed by JavaScript and C#. However, we don't find any details about job market for GW Basic and Pascal.

3.7   Community Support

The community support for each of the above language is available. Comprehensive documentation for Java is available in the form of java docs. The community support for Java is available at. Similarly support for C# is available at. JavaScript is an emerging language and support for AJAX, j-Query, Angular Js is available at different forums such as. Even though, support for Pascal, GW Basic and C language is available in the forms of books, online courses, and some forums, but their support is not comprehensive.

3.8   OS/Machine Limitations

The IDE for GW Basic, C and Pascal can easily be run with low configuration requirements such as on Pentium IV. JavaScript doesn't require any specific platform to run as it can be easily run on a browser. Java Development Kit (JDK) can be run on small configuration, however, its IDE requires Core 2 Duo machine to run. The latest version of Microsoft Visual Studio requires Windows 8 to run.

3.9 Extensions/ Library available

The extensions for Java and C# are easily available for general purpose task such as web programming, Bluetooth communication, hardware interfacing and Graphical User Interface (GUI). This includes Servlets, Java Server Pages (JSP), Asynchronous JavaScript and XML (AJAX) toolkit etc. JavaScript is an emerging language and a large number of extensions for it is available such as Angular-JS, Expres-Js, Socket.IO, Node-Js, D3, Fire Base and Web-GL etc. C language has been used for many years; hence, extensions for it are readily available such as OpenGL, NS2 extensions etc.

3.10   Coverage

For a primary language to be taught to students, it is essential that it extensively covers various notions such as OOP, multithreading, databases, networking and mobile computing etc. An introductory programming language once learnt by student can be used primarily for demonstrating these advanced notions of computer science. This section discusses how the major programming languages provide support for these concepts.

3.10.1 Object Oriented Programming

The languages like C, GW Basic and Pascal don't provide support for object oriented programming. C++, an extension of C language can be used for object oriented programming. Both Java and C# are 100% object oriented programming languages. It has support for implementing concepts such as encapsulation, inheritance, polymorphism, aggregation and composition. JavaScript can be regarded as an object

based programming language as it has support for basic concepts of object oriented programming such as classes, association, and inheritance. However, concepts such as encapsulation and polymorphism are not supported. Inheritance can be implemented using prototypes.

```
var student = {
  id: 2,
calmarks: function(b){
return marks;
  }
};

var graduateStudent = Object.create(student);
p.dissertationTitle = "Data Mining";
```

Figure 5: Inheritance in JavaScript

3.11   Databases

The concepts of database can be implemented in C# and Java using ADO .NET and JDBC. In addition, database connectivity can be provided for document databases such as MongoDB. JavaScript can be used to connect to any type of databases such as MySQL, SQL Server, MongoDB etc. In C language, database connection can be made using ODBC API. Interfacing with databases is a very trivial task in GW Basic and Pascal language.

3.12   Operating System

C, C# and Java provide support for implementing different operating system concepts such as multithreading, I/O, monitor, semaphores, memory management and process management. In JavaScript, the concept of threads is slightly different. Worker threads are used to implement multithreading. Figure 6 shows the example. File I/O can be performed but in a sandboxed environment. Performing multithreading in Pascal and GW Basic is not possible.

```
var w = new Worker("thread2.js");
w.postMessage("Hello");

w.terminate();
```

Figure 6: Multithreading in JavaScript using Worker threads

3.13   Low Level Programming

In Pascal, C language low level programming task can be easily performed such as using interrupts, process management, inter-process communication, protected operating system calls etc. In C#, these operations can be performed in unsafe region. In Java, system level programming can be done using native methods. GW Basic can interface with assembly language by using USR function and CALL statement.

```
uses Dos;
        procedure BORDER (color : byte);
        var regs : registers;
        begin
        FillChar (regs, SizeOf(regs), 0);
         regs.ax := $0B00;
         regs.bh := $00;
         regs.bl := color;
         Intr ($10, regs);
end;
```

Figure 7: Using interrupts in Pascal

3.14   Network programming

Pascal, C, Java and C# provide support for network programming using Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) sockets. Figure 8 shows an example of TCP sockets in Java. In JavaScript, socket communication can be performed using Socket.IO API. Socket programming in Pascal and GW Basic is not a trivial task.

```
Import java.net.*;
…
try {
   Socket s = new Socket(host,port);
   OutputStream os = s.getOutputStream();
   os.write("Data");
   os.

close();
}catch(Exception e) {
   System.out.println("Error");
}
```

Figure 8: Socket programming in Java

3.15   Web programming

In Java, web programming can be done using Servlets, Java Server Pagse (JSP), Struts and Java Server Faces (JSF). In C#, Active Server Pages (ASP) .NET can be used for web programming. JavaScript can be used for both client side and server side programming. At server side, this can be done using Express-Js. At client side different APIs such as j-Query, Angular-Js, Bootstrap etc. In C and Pascal language, web programming can be done using Common Gateway Interface (CGI) programming. Having said that, it should be noted that programming in CGI is a very cumbersome job and that makes web programming in C and Pascal very difficult job.

3.16   Mobile Computing

Java provides support for mobile computing using Android and J2ME etc. In C#, XARMIN can be used for cross platform mobile application development. In JavaScript, mobile application development can be done using Phone Gap API or Win Js platform. Pascal and GW Basic is not popular for mobile application development.

4. RESULTS & DISCUSSIONS

Based on the above discussion, it can be concluded that Java is the best general purpose programming languages to be used for teaching computer science concepts. It has good writability, reliability, market demand and can be used to teach any computer science concept such as operating system, mobile computing etc. Besides Java, C# can also be used to teach computer programming. Languages such as Pascal and GW Basic had been used widely to teach introductory level course, but are no longer in demand in market, nor can they be used to implement modern concepts of computer science such as delegates, design patterns and object oriented programming etc.

CONCLUSION

In this paper, an analysis of major programming languages of computer science is done. The paper compares the selected languages based on different factors such as their readability, writability, support, market demand and coverage. It has been concluded that Java is the most appropriate language to be used for teaching computer science concepts.

ACKNOWLEDGMENT

REFERENCES

Robins, A., J. Rountree, and N. Rooftree . (2003) Learning and Teaching Programming: A Review and Discussion. Computer Science Education, vol. 13(2), Springer, p. 137-172.

MILNE, I. and G. ROWE.(2002) Difficulties in Learning and Teaching Programming—Views of Students and Tutors. Education and Information Technologies, vol. 7(1), China, p. 55-60.

Davies, S., J.A. PolackWahl, and K. Anewalt . (2011) A Snapshot of Current Practices in Teaching the Introductory Programming Sequence.beligin, pp.23.

Schulte, C. and J. Bennedsen. (2009), What do teachers teach in introductory programming?, IEEE conference proceeing,pp.58-65.

Mason, R., G. Cooper, and M.D. Raadt.(2012) Trends in Introductory Programming Courses in Australian Universities – Languages, Environments and Pedagogy. in Proceedings of the Fourteenth Australasian Computing Education Conference (ACE2012), Melbourne, Australia.pp.89.

Sebesta, R.W. (2014) Concepts of Programming Languages, USA, pp.60.

Moritz Schlarb , Christian Hundt and Bertil Schmidt.(2015) SAUCE: A Web-Based Automated Assessment Tool for Teaching Parallel Programming", Euro-Par 2015: Parallel Processing Workshops.

Nikolai Tillmann, Michal Moskal, Jonathan de Halleux, Manuel Fahndrich, Judith Bishop, Arjmand Samuel, Tao Xie, July 3.(2012) The Future of Teaching Programming is on Mobile Devices, ITiCSE12 , Haifa, Israel,pp.25.

Lisa Eadicicco. Dec 6 (2014) The Best Programming Languages Every Beginner Should Learn, Business Insider, pp.82-85.

Rebecca Hiscott. Jan 21(2014) Programming Languages You Should Learn Right Now, Mashable, USA, pp.57.