Why C# and Why .NET In The Undergraduate Information Systems Curriculum

Mehdi Raoufi
Department of Computer Science and Information Systems
Southwestern Oklahoma State University
Weatherford, OK 73098

and

John Maniotes
Department of Computer Information Systems
and Information Technology
Purdue University Calumet
Hammond, IN 46323

Abstract

Considering the rapid pace of changes in the software field and the limited courses that a student can take in languages, the question is which languages are crucial for students to learn in an undergraduate IS curriculum. This paper investigates the necessity of teaching C# and .NET in the undergraduate IS curriculum. It explores the pros and cons of .Net versus J2EE for applications development and differences between C#, C++ and Java, and which one may be the best language for teaching first programming course in IS curriculum.

Keywords: C#, .NET, framework, platform, Common Language Runtime (CLR), Intermediate Language (IL), undergraduate IS curriculum

1. INTRODUCTION

C# and .NET

C# is Microsoft's latest object-oriented programming language developed for .NET platform and .NET is Microsoft's latest platform technology for creating web services. C# is a C++ based language and was developed to provide portability for distributed applications over network and internet. Application development in .NET platform can be done in multiple languages including C#, C++, and Visual Basic. Programs developed in all of these languages are compiled to Microsoft's Intermediate Language (IL) and executed within Common Language Runtime (CLR). We explain the core elements of .NET and how web applications are developed and run with this technology. .NET is not a programming language; it's a virtual-machine technology (similar to Java virtual machine technology) with a framework that provides capability to run a variety of web applications. The .NET framework class library provides a set of classes that provide essential functionality for applications build within the .NET environment. Web functionality, XML support, database support, threading and distributed computing support is provided by the .NET framework class library. All .NET code is translated to Microsoft Intermediate Language (MSIL) and run within CLR; CLR is similar to Java Virtual Machine (JVM). The IL code is language-independent and similar to the Java byte code. A single .NET application may consist of several different languages. Two very important features of CLR are language interoperability and language independence.

ASP and .NET

ASP.NET is an upgraded version of Active Server Pages (ASP). Developers can create COM objects in Visual Basic, Visual C++ or C# and then use ASP.NET page to call on those objects for a web interface. .NET supports the existing Web services standards like XML, SOAP (Simple Object Access Protocol). Hence, a Web service remains as an open technology regardless of what platform is used .NET or J2EE. For example, a J2EE program can access information from .NET if proper standard is followed for information retrieval and vise versa. .NET managed components are used to build the business layer of the .NET application. The basic function of this layer includes performing business processing and data logic. It connects with databases using Active Data Objects (ADO).

2. PROS AND CONS OF .NET VS J2EE

It is expected that both .NET and J2EE will have large market for enterprise web applications and internet services. We express some advantages and disadvantages of each framework. Developments in J2EE are restricted to Java and developments in .NET are restricted to Windows. Sun Microsystems has gained the entire industry including IBM and Oracle behind its J2EE standard. On the other hand, .NET is based on Microsoft's efforts to gain the entire Web services market share. Both .NET and J2EE offer rapid application deployment features. Some are more powerful in .NET compared with J2EE and some are more powerful in J2EE. The real advantage of ASP.NET is that it is independent of client device. Microsoft .NET offers Components and E-commerce capabilities which in many cases are superior to features found in J2EE. J2EE offers solutions from multiple vendors with a wide variety of tools often times are not interoperable because of portability problems, but Java Runtime Environment (JRE) is available on any platform including Win 32 and UNIX; J2EE offers easy and effective portability.

3. C# AS A FIRST PROGRAMMING LANGUAGE IN IS CURRICULUM

C#, Java, and C++

Both C# and Java improve on C++ in a similar ways, so we are not going to explain features like the benefits of a single inheritance or other features in C# and Java over C++. We will be explaining summary of the differences between C# and Java, and then we will explore the new features of C# and how using C# may improve teaching first object-oriented programming course in Information Systems curriculum.

The C# language was built with the observation of many languages, but mostly Java and C++. C# boasts type-safety, garbage collection, simplified type declarations, versioning among other features that make developing solutions faster and easier, especially for COM+ and Web services.

C# Differences with Java

C# has several important new features that make us to distinguish it from Java. Some of these features, such as delegates and events, are very useful and provide functionality built into the language that, in Java, requires coding and a rigorous understanding of Java design patterns. This section covers important features of C#.

Primitive Data Types: Java has quite a few primitive data types: byte, char, int, long, floats, double. This is not the case in C#. C# uses the .NET object/type system so that C# programs can communicate with other .NET languages without having type confusion. This enables the primitive, or simple, types in C# function just like any other object.

Statement Comparisons: Statements in C# and Java are very similar, since both languages descend primarily from C and C++. The difference between keywords “import” in Java and “using” in C# is that Java has a packages concept, while C# uses namespaces similar to those of C++. The keyword “using” makes all names in the given namespace accessible to a program.

Conditional Statements: C# has both of the structures "if-then-else" and "switch". Both are similar except for one difference in the C# "switch" statement syntax. Java allows for control flow to implicitly fall between different cases in the switch statement, whereas the C# compiler explicitly does not allow this, the C# compiler will mark this as a syntactical error.

Definition of Class: Definition of Class and inheritance in C# is similar to Java except that in C# classes are inherited from the System.Object class, just as all classes in Java are inherited from the java.lang.Object class. Defining a class which is inherited from another class would be written using ":" instead of the "extends" keyword.

Indexers: A class's indexer lets you access any instance of that class as if it were an array. A class may define multiple indexers, each of which differs by the number and type of its arguments. Indexers are very similar to properties, especially in the syntax for defining them.

Structs: A struct is similar to struct in C++, except that a C# struct can have any kind of class member, including constructors and methods; and the default accessibility for struct members in C# is private, rather than public as in C++. Similar to C++ structs, C# structs always copy by value and are therefore both mutable and exempt from dynamic memory management (but, memory link).

Enums: Another class member type Java does not provide is the enum. While similar to enums in C++, C# enums are based on an "underlying type," which can be any signed or unsigned integer type. Enumerations are derived from the built-in class System.Enum, and therefore every enum inherits all of that class's members.

Properties: The concept of using "getter" methods to encapsulate internal object properties is a design pattern that spans object-oriented languages. It isn't limited to C# or Java. C# has taken the concept of properties a step further by actually building "getter" methods into the language semantics. An object property has a type, a set method, and a get method. The set and get methods of the property determine how the property's value is set and retrieved.

Delegates: Delegates are C#'s answer to C++ function pointers; except that delegates are safe, and function pointers are dangerous. Both Java and C# have removed the function pointers, finding safer ways to maintain references to behavior that is determined at runtime.

Events and event notification: C# events operate very much like Java events, except that C#'s are integrated into the language. For a class to receive an event, it must have a field or property that is a delegate marked with the event keyword. From inside the class, this delegate member is just like any other delegate; it can be called, checked to see if it's null, and so forth. From outside the class, there are only two things you can do with it; add or remove a method to the delegate, using the operator+= or the operator-=, respectively.

Operator overloading: C# permits operator overloading, using syntax almost identical to that of C++. Some operators that can be overloaded in C++, such as operator=, cannot be overloaded in C#. Java's creators decided to leave operator overloading out of Java.

Methods: C# methods have some features that Java does not. In particular, C# provides several modifiers on method parameters and has keywords for virtual methods and method overriding. As in C++, C# method parameters are value parameters. The ref modifier on a method parameter makes that parameter a reference. The out modifier indicates that the parameter is an output parameter, which is identical to a reference parameter, except that the parameter must be assigned before the method's returns statement. C# methods are, by default, no virtual but can be made virtual by explicit use of the keyword virtual.

Teaching first programming course using C#: Java has several weaknesses as introductory level language. It forces beginning students to learn several sophisticated concepts even to write a very simple program. Students who are new to programming should sort through some complicated issues before learning the main notion. This becomes very frustrated with some students who discard programming without giving it a good chance to succeed. One of the difficulties of Java for beginning students is standard input; input operations in Java are very complicated for beginning students to understand. To read a simple integer or a real value a student should use a large number of codes, which include while loop, Boolean variable, string operations, wrapper classes and exceptions. Another difficulty with Java is exception handling; Java requires every method to catch the exception generated in the body of program or verify the possibility of these exceptions in a "throws" clause. Hence, Java programmers should learn and use exception for a simple program. We believe that using C# as first programming language overcomes these and other difficulties associated with Java.

4. CONCLUSION

Authors believe that C# and Microsoft Visual Studio .NET provides a more proper language and environment to teach introductory course for beginning students. We also believe that the Java platform is neutral with respect to operating systems. If you don't like the services of a vendor, you can find another vendor that meets your requirements. It will take some time before that can be said about C# or .Net. While C# is a fine language and is a necessary language for Information Systems students to learn for Windows application, it will be able to compete with Java when C# is freed from its Windows dependence. On the other hand, we cannot ignore the tremendous financial and technical resources that Microsoft has. Microsoft is the dominate force on the PC Windows operating system. For now we think that Microsoft's dominance and financial muscle may propels C# over Java in the IT community. Regardless of which language Java or C# triumphs in the IT community, the authors recommend C# as a necessary language in the IS curriculum for students who are interested in application development and web services.

5. REFERENCES

Deitel, H. M. and P. J. Deitel, 2002, C#: A Programmer's Introduction, Prentice Hall, Upper Saddle River, NJ.

Gittleman, Art, 2003, Computing with C# and .NET Framework, Jones and Bartlett

Gittleman, Art, 2005, C# .NET Illuminated, Jones and Bartlett

Lippman, S. B., 2001, C# Primer: A Practical Approach, Addison Wesley, Reading, MA.

Schildt, H., 2001, C# : A Beginner's Guide, Osborne McGraw-Hill, New York, NY.

Sharp, J. and J. Jagger, 2002, Microsoft Visual C#. Net Step by Step, Microsoft Press.

Watson, K., et al. 2002, Beginning Visual C#, J. Wiley & Sons, New York, NY.