

INTRODUCTION

There is a lot of confusion about which programming language to choose when introducing students to programming and programming principles. Many teachers tasked with teaching Computing or Computer science tend to be ICT teachers who have morphed into computing teachers, pushed long by the governments drive to make computing a replacement for ICT as a subject. Many of these teachers do not have a background in programming and any programming they may have done was years ago in an environment which would be alien to today's students.

The main options available at the moment for teachers and therefore students are to be Python, C#, C++ or Visual basic. With many Schools and Colleges opting for Python.

The aim of this document is to hopefully provide a little more clarity and help teachers and institutions make an informed choice of which language to use for their students learning, coursework and preparation for the future.

It is also worth mentioning from a computer science teacher's point of view, the choice of language is pivotal in the engagement reciprocated by the learners when learning what is in effect a new way of communicating.

PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python is also seen as a scripting language like PHP, Perl, Ruby. Often used for rapid solution development to solve problems. It can also be used for web programming (Zope, Google App Engine, and much more). Being an interpreted programming language it does not lend itself to creating stand-alone executable programs which can be shared as single programs, runnable by clicking. None technical people would struggle to use a Python program without understanding the use of the interpreter software.

Python has risen to fame relatively recently with many educationalists and programmers being of the opinion that Python should be first programming language to be learnt. The argument being; you will quickly learn how to think like a programmer. Python is very readable. You won't waste a lot of time memorizing the arcane syntax that other programming languages will present you. Instead, you will be able to focus on learning programming concepts and paradigms. Once you have those tools under your belt you can move on to other more powerful, specific languages and readily understand a given piece of code. Python is super-powerful! There's a reason NASA uses it.

Python is easy to learn. The learning curve is very gradual. Other languages can be quite steep. As I mentioned above, Python places an emphasis on readability. Here's an example of how easy and readable Python is. The first code example below is written in C:

| Some code in C | Code with the same output in Python 3 |
|---|--|
| <pre>#include <stdio.h> int main() { printf("Hello, world!\n"); }</pre> | <pre>print("Hello, world!")</pre> |

Python gives a lot of flexibility in fulfilling tasks because of its “relaxed nature”, as many programming principles are not as rigorously enforced within the language. The down side to this being it does not enforce the learning of strict programming principles and syntax structures required by other languages. This makes transition to other languages difficult for some.

As a first language it runs the risk of teaching, what would be considered ‘bad practice’ in the programming world and does not cover the standard syntax used in many other widely used programming languages.

The development environments available for Python also tend to be very basic and often consist of a text editor with an interpreter bolted on to it. This can be very difficult for new users to use as development environments as they lack features such a dynamic syntax and error checking. Quite often an inexperienced programmer will be given a long list of errors when a program is first run.

Python does not easily provide the ability for the programmer to create graphical user interfaces.

Teaching programming principles and introducing programming concepts to none programmers can be very frustrating and troublesome with such a language and limited development environment.

VISUAL BASIC

Visual Basic is a third generation event-driven programming language. The Microsoft Corporation released Visual Basic in 1987. It was indeed the first visual development tool from Microsoft. Visual Basic was derived from BASIC and enables rapid application development of graphical user interface applications, access to databases using controls and objects. The language not only allows Visual Basic programmers to create simple GUI applications, but also helps them develop quite complex applications. Visual Basic allows developers to target Windows, Web, and mobile devices.

Programming in Visual Basic is a combination of visually arranging components on a form, specifying attributes and actions of those components. Since the default attributes and actions ought to be defined for the components, it is very simple to write a program without the help of a Visual Basic programmer. Forms can be created using drag and drop techniques. Visual Basic provides many interesting sets of tools to help you in building exciting and entertaining applications. It provides these tools to make your life easier, since the entire coding is already written for you. Moreover, it is a user friendly language which is very effective and efficient. A tool is used to place controls such as text boxes, buttons, etc on the form window. Default values will be provided when a control is created, but it can be changed by the Visual Basic programmer.

Visual Basic is not only a programming language, but it also has a complete graphical development environment. Visual Basic has the ability to develop programs that can be used as a front end application to a database system, and serving as the user interface which collects input from the user and displays formatted output in an attractive format. As the Visual Basic programmer works in the graphical environment, much of the program code is automatically generated by the Visual Basic program. Visual Basic can create executables i.e. EXE files, ActiveX controls, but it is primarily used to develop Windows applications. It is also used to interface web database systems. This generation of Visual Basic continues the tradition of giving you a faster and easier way to create .NET framework-based applications. Visual Basic also fully integrates the .NET framework and the common language runtime, which provide language interoperability, garbage collection, enhanced security, and versioning support.

Although Visual basic is not used extensively in areas such as the games industry due to its limitations in real time applications and processing times. It is still used within industry for developing single standalone interactive systems and applications. The language is closer to written English and a result easily readable and code can be interpreted easily. It is still considered as one of the best structured high level programming languages most suited to learning and practicing structured programming and learning programming principles and concepts. Concepts which can then be carried forward to learn most other programming languages based on those foundations and principles. Visual basic is not used at university level on Computing courses as it is not considered be used enough in industry and therefore not suited to a degree level computing course. Visual studio has developed as a very user friendly and aesthetically pleasing

develop environment with many user friendly debugging and informative tools to develop the programmers skill levels. This lends its self to “playing around” with the software allowing the user to learn through exploration without the risk of serious damage to the computer or the software. It also allows very quick event driven program development using drag and drop interactivity.

C# / C++

These development languages are extensively used in industry and are considered professional programming languages. They are extremely powerful languages overcoming the performance issues linked to Visual basic. Both C# and C++ are based on the same programming principles and look very similar in structure and syntax. However, the languages are much further removed from English and therefore much more difficult to read, decipher and debug due to their complexity. This presents problems for inexperienced programmers. There are also other complexities such as case sensitivity and declaration methods which would present problems to inexperienced programmers.

Here’s an example of how easy and readable Visual Basic is compared to C#.

| Visual Basic Code | C# Code |
|--|---|
| Imports System | using System; |
| Namespace Hello | namespace Hello { |
| Class HelloWorld | public class HelloWorld { |
| Overloads Shared Sub Main(ByVal args() As String) | public static void Main(string[] args) { |
| Dim name As String = "VB.NET" | string name = "C#"; |
| | // See if an argument was passed from the command |
| 'See if an argument was passed from the command line | line |
| If args.Length = 1 Then name = args(0) | if (args.Length == 1) |
| | name = args[0]; |
| Console.WriteLine("Hello, " & name & "!") | Console.WriteLine("Hello, " + name + "!"); |
| End Sub | } |
| End Class | } |
| End Namespace | } |

LEARNING AND TEACHING A PROGRAMMING LANGUAGE

Very few ICT teachers have any prior structured programming experience. As a result they need to be taught the principles of programming and development in a programming language within a very short period of time. In turn they need to convey their knowledge to the students in a simple way in a short time to get them up and running and independently programming. This is especially difficult with the GCSE and A level students who need to reach a level of competency before they start their programming project work late in the autumn term. This requires a very steep learning curve and considerable commitment on the part of the students, as learning programming is just like learning any other language and requires practice.

INDUSTRY STANDARDS

Many would argue that the languages used in industry should be the ones taught at school and college. This however has three major drawbacks.

Firstly, the languages used in industry are used by professionals usually developing for niche software markets usually with an emphasis on concerns such as production time, security and performance. The structure and the method of programming is therefore not an issue. Where as in teaching the emphasis needs to be on learning the fundamental structures, conventions and a focus on assessment of the quality of the product according to assessment criteria. Teaching a programming language without the constructs and foundations would mean the student would not be able

to transfer those skills to the learning of any other language. It would be like teaching French words without the grammatical sentence constructs.

Secondly, the depth to which the students use a programming language within a school or college course is far less than required for industrial work. Therefore the student would not be adequately prepared for industrial standard work anyway.

Thirdly, with the nature of programming and the IT industry in general, with the very rapid developments of programming languages and newer versions of the same language being released all the time. As with technologies such as Flash and other specialist software the practical application lifespan is very limited. The specifics of a language learnt today may be out of date within a couple of years. A perfect example is mobile App development where if a student learnt the technologies three years ago relating to this area. These technologies and tools are now be obsolete.

EXAM BOARD OR COURSE REQUIREMENTS

Most exam boards are not prescriptive in what programming language should be use for project work. The subject specifications contain programming principles which can be demonstrated and explained using a range of languages. The exam board will normally recommend a number of preferred languages which meet the requirements of the course. Most exam boards will allow coursework created in Python, C, Java or Visual Basic.**

** check your exam boards latest specification for recommendations.

MORE READING

Choice of Programming Language – Justifying my Choice. - <https://mrgict.wordpress.com/2015/01/13/choice-of-programming-language-justifying-my-choice/>

Python for Visual Basic programmers -<https://github.com/raspberrypilearning/python-for-vb-programmers/blob/master/worksheet.md>