

A Vision in Teaching Programming

(UAE Higher Education As A Case Study)

Murad Al-Rajab¹, Samia Loucif^d, Mohamed Sobhi Toubar¹

¹Software Engineering Department, ALHOSN University
Abu Dhabi, United Arab Emirates

¹m.alrajab@alhosnu.ae

¹s.loucif@alhosnu.ae

¹mstoubar01@students.alhosnu.ae

Abstract— Teaching and studying programming is a big challenge for both teachers and students alike. For teachers it is always the matter of explaining to students how to explore the programming world with all its dimensions like syntax, coding, algorithms, etc in order to view it clearly and in the correct manner, noting that computer programs are part of our daily life. On the other hand students struggle when they study programming; either they are studying it for the first time or those who have an adequate programming background.

In this paper we will focus on programming and on our experience (vision) in teaching programming at ALHOSN University in UAE, programming languages that are taught in UAE higher education institutes, the methodologies we are following in teaching programming for novice students, the main problems they encounter in learning programming and the problems solving techniques we adopt at ALHOSN University in order to make programming more fun than just passing a course. Also the paper will emphasize the tripod of programming (Thinking, Writing Algorithms, and then Programming), in a more scientific methodology.

Keywords— Programming, Programming Languages, Problem Solving Techniques, Teaching, Methodology, learning programming

I. INTRODUCTION

Programming is the process of building computer software or in other words it is the procedure of building computer applications and is defined to be a list of instructions to the computer in order to operate. We are all surrounded with computer programs, which actually assist us in doing and accomplishing our daily tasks. Due to that, teaching programming as a course is being a core in multiple disciplines like Computer Science, Computer Engineering, Information Systems, Management Information Systems, Industrial Engineering, Electrical and Telecommunications Engineering, Mechanical Engineering, Bioinformatics, and more other fields.

Computer software or programs are found in almost all fields in the daily life; so teaching such a course is being mandatory in different departments as noticed. In teaching programming; four concepts must be recognized in order to understand the key of the field; “Program, programming, programming languages and programmer”.

The main clue of this paper is to set the role of each concept mentioned above in a more scientific way. The paper will discuss programming languages origin, programming languages used in teaching in UAE higher education, teaching programming by experience at ALHOSN University, the methodologies used, problem solving techniques followed and how to improve teaching programming and make programming sessions more fun and will conclude by the future techniques used in studying programming.

The paper is organized as follows. Section 2 reviews related work. Section 3 will discuss a brief about programming languages. While in section 4, a summary of the main high level programming languages functionality will be listed. Moreover, a list of programming languages used in teaching at UAE higher education will be presented in section 5, while Section 6 points the problems faced in teaching programming in the higher education, On the other hand, Section 7 addresses our experience in teaching programming at ALHOSN University and ways to improve out the course, and finally section 8 concludes the paper.

II. BACKGROUND AND RELATED WORK

Many studies have been conducted to improve the way programming courses are taught and many researches were carried out in order to investigate how to make teaching and studying programming much easier. The authors in [1] discussed that in order to help students on how to write algorithms and solve programming problems easily, it would be better to use a new tool that will increase their knowledge and skills in that area, and suggested to use Alice programming language as a tool because it supports 3D environment. Trevor Nesbit [2] has pointed out that there are two aspects and issues that must be considered prior to teaching introductory programming courses, the first is related to the methodology of programming either procedural or OOP, the second is the analysis and design concepts that should be implemented before coding or teaching coding. In [3], Hu has implemented a case study to teach adults programming language, by stating out that for teaching programming first a proper programming language, proper environment, and proper book must be chosen carefully in order to meet students’ styles and cultural backgrounds. In [4], authors discuss a new technique in solving the challenges of teach programming. For example they suggest teaching students Excel first to understand arrays, built

in functions, input and output, HTML for program layout and MATLAB to understand loops; thus introducing new teaching concepts for easier programming. While in [5], authors suggest teaching introductory programming courses to students, how to program games through a game engine, which will allow the students to develop the fundamental programming concepts and skills without having to learn the syntax.

On the other hand, authors in [6], tried to solve the issue of spending time in teaching the syntax more than the time spent in development and problem solving. So they solved this issue by newly customized software as a tool for a first year programming course MIS students. Matthiasdottir in [7], introduced a new program to teach novice programmer programming without lectures but by computer lab sessions, recorded lectures and online exams. In his solution each student has two screens, one to watch the recorded lecture and the other to practice programming code. But authors in [8], address an online tutoring system to teach students different programming languages and allow teachers to exchange ideas.

Furthermore, the authors in [9], studied the difficulties in teaching and learning programming based on a survey over 500 students who are studying programming as an introductory course for different majors. Their study will help in developing materials for learning and teaching programming courses based on the difficulties studied.

III. PROGRAMMING LANGUAES

To become familiar with programming languages, you must go back to the origin and evolution of these languages to get a clear understanding of their development. It is better to teach students how languages have evolved to what we have today, but it is not easy to teach students history of programming languages as they don't like historical lectures [10], So, teaching a brief historical introduction to programming languages and focus only on the generations of these languages will open the doors for students to be interested in the new techniques of these languages. In general, Programming Languages can be divided into five main generations as illustrated by "Fig 1" below:

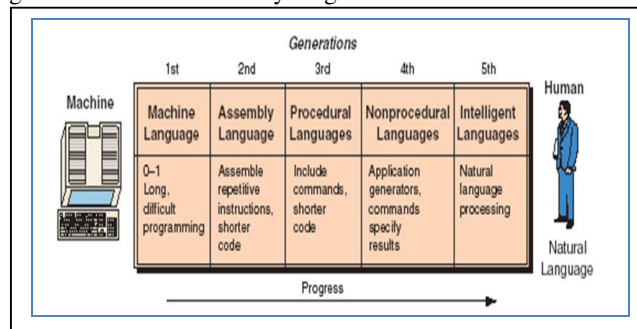


Figure 1: Programming Languages Generations

IV. HIGH LEVEL PROGRAMING LANGUAES FUNCTIONALITY

There are many high level programming languages available nowadays, but it is not easy to decide which programming language to teach in higher education. Here we list the top 10

high level programming languages in 2012 and their common use, this statistics is based on OMG top ten website [11], the order can be changed but the list of the top 10 programming languages is still there based on multiple other websites:

TABLE I. TOP 10 HIGH LEVEL PROGRAMMING LANGUAES

TOP 10 HIGH LEVEL PROGRAMMING LANGUAES	
Programming Language	Function/Use
Java	It is a general purpose programming language that is used in everything from desktop software, to web applications and services. Java is a powerful, portable object oriented (OO) programming language. It's used on Web pages, in cell phones and on mainframes all around the world. A program written in Java on a PC can easily find its way into many different computing platforms. This makes Java a popular language for application development.
C++	It is a general purpose programming languages, with application domains including systems software, application software, server and client applications, and entertainment software such as video games.
C#	Is a general-purpose programming language. Is used to create both Web-Based and GUI-Based applications on Windows ranging from computer games, utilities, Operating Systems and compilers.
PHP	Is especially suited for Web development because of it easy embedding into HTML pages. It is an open-source, server-side, cross-platform, interpretive HTML scripting language
Ruby	Is used in typical scripting language applications such as text processing and middleware programs, and is also suitable for larger software systems. Its most successful application is in the Ruby on Rails web framework
Python	Is a general-purpose, high-level programming language. Python is used in many application domains such as Web and Internet development, Database access, Desktop GUIs, Scientific and Numeric Applications, Educational Programs, Network Programming, Games and 3D Graphics.
C	A general purpose programming language, Although C is a relatively

TOP 10 HIGH LEVEL PROGRAMMING LANGUAGES	
Programming Language	Function/Use
	old language; it is still widely used for system programming, writing other programming languages, and in embedded systems.
Visual Basic	Is used to develop client applications, Web apps, script, and macros embedded in other applications such as Microsoft Excel.
Java Script	It is used in millions of web pages to authenticate forms, detect browsers and improve design, and it is easier to run these functions as it is embedded into HTML.
Perl	is used to develop data analysis, web applications, text manipulation, batch processing, Bio-informatics, Games, Network and command-line utilities; and these are just a few of the basic features. Also It is used extensively to write Web server programs for a variety of tasks.

V. A VIEW OF THE HIGH LEVEL PROGRAMMING LANGUAGES IN UAE HIGHER EDUCATION

Among the multiple different programming languages used all over the world and the rapid challenge that is used to select the best one, we did a statistical chart showing out the most used programming languages among different UAE higher education institutes. Around 20 Universities are included in the statistics as shown in “Fig 2” below:

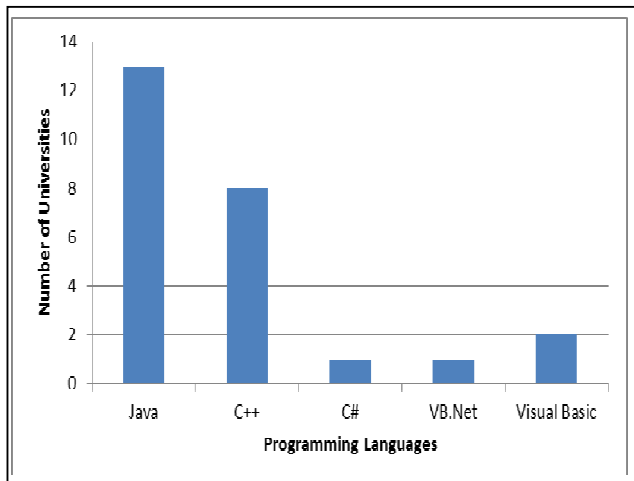


Figure 2: Programming Language Statistics used in UAE higher education institutes

From the figure we can notice that around 13 Universities teach Java, 8 teach C++ and 2 or less universities teach C#, VB.Net and Visual Basic. Also it was noticed that some Universities teach more than one programming language in

the same curriculum in Computer major programs or engineering fields.

Java is the number one programming language used in most UAE higher education institutes because it is used on Web pages, in cell phones and on mainframes all around the world. A program written in Java on a PC can easily find its way into many different computing platforms. This makes Java a popular language for application development, in addition its syntax gives more logic to educators.

VI. PROBLEMS IN TEACHING PROGRAMMING IN HIGHER EDUCATION

Higher education here means Universities and colleges. From the literature review, it was clear that there were and still are many contributions to develop programming teaching methodologies, and all that due to difficulties faced in learning programming for a novice programmer. The paper will state some of the important problems and challenges encountered by both learners (students) and instructors while conducting programming course, a survey was done and distributed to both students and programming instructors in different universities and institutes among UAE, and summary of the main problems faced by them was collected in table 2 below:

TABLE II. PROBLEMS OF LEARNING AND TEACHING PROGRAMMING

Problems of learning and teaching programming	
Problems Faced By Learners (students)	Problems Faced by Instructors
<ul style="list-style-type: none"> •The quality of the course is new, since students have no earlier background in it like other courses such as math or physics. •Becoming familiar with the syntax of the programming language as it is something new to their knowledge. •Linking between theoretical concepts and practical sessions. •The awareness that students are learning a new language. •Students don't know how to study the course and deal with it. •The continuity of the course and subjects are related. •Some students are not familiar with how computers operate. •Students are not familiar with reading the programmatic problem and analyzing it. •Students are not interested to read and fix programming syntax errors 	<ul style="list-style-type: none"> •Timing is the big issue for the introductory programming course to be covered in one semester •The detailed points in programming syntax and when to explain them. •Students' ability to think, link and program. •Classroom facilities •Selecting the right book. •Selecting the right tools •Selecting the appropriate language based on the field of study. •Selecting the appropriate methodology of programming to teach either procedural or OOP. •Selecting the right techniques that may help understanding programming concepts (like loops, arrays, ...) •A

VII. EXPERIENCE IN TEACHING PROGRAMMING COURSES AT ALHOSN UNIVERSITY

ALHOSN University is a private University in the United Arab Emirates which offers programming courses under the umbrella of multiple departments, Software Engineering, Civil Engineering, Industrial Engineering, Mechanical Engineering

and Management Information Systems. It is noticed that ALHOSN University is offering the course under two faculties; faculty of Engineering and Applied Sciences and Faculty of Business.

The paper will highlight our experience in teaching programming for multiple engineering discipline students (Software, Civil, Industrial, and Mechanical). Also we will point some advisable points to students on how to study programming and deal with it as a daily language.

All engineering students are required to register in FES 111 Programming I, as it is mandatory for the Software Engineering students to register the course in their first year semester but for other engineering students, they are advised to register the course also in their first year semester, but they can postpone it to later semesters. There are two other mandatory Programming courses to be taken by Software and Industrial Engineering students that are based on FES 111 Programming I. It was noticed that around 50% of engineering students prefer to postpone taking programming I course or they struggle when they study the course and that was a big challenge faced by the programming teachers at ALHOSN University to solve this issue. Basically FES 111 Programming I; is offered from the Software Engineering Department, and all Instructors who taught the course are from computer science background and have experience in teaching programming languages.

Teaching programming is a story full of challenge and motivation at the same time to the software engineering teachers and to the engineering students, the main aim that we target at ALHOSN University by teaching students programming is to deliver a new motivated generation of programmers who can learn a new daily used language in a semester and use its alphabets and syntax to communicate with the computer world.

Java Programming Language is the language used in FES 111 Programming I course, and JCreator Light Edition [12] is the IDE used, our course outline plan is divided into 16 weeks, the number of weeks can be adjusted to 15 according to the semester circumstances.

It is noticed that Java was selected to be the language of coding the fundamental programming and that because java has many advantages and is a useful language for coding engineering applications, add to that the syntax gives more logic to learners and not so much difficult, as well as it is a free language and machine independent.

JCreator LE is the IDE used by students in Programming I, since it has a simple graphical user interface, light to launch and easy to write code in, compile code and execute programs. After around 6 years, 18 semesters offering the course it was adjusted to teach around 6 to 7 chapters in Programming I based on the semester timings. Starting from introduction to Java up to Arrays and/ or OOP basics. The FES 111 is scheduled to be 3 credit hours lecture course that is divided weekly into 3 hours lectures broken to two theoretical classes per week, 3 hours laboratory sessions per week and 1 hour, 30 minutes Tutorial session. The course has one pre requisite which is school Math grade 12.

“Fig 3” illustrates our key in introducing the course to students in the first week. Because students are not familiar with programming, it is mandatory to explain four basic concepts and make sure that they understand the difference in between. The four concepts are “programmer, programming, programming language, and program”. It must be clear that that programmer is the person who will use programming languages as a tool through a programming process to write and develop programs.

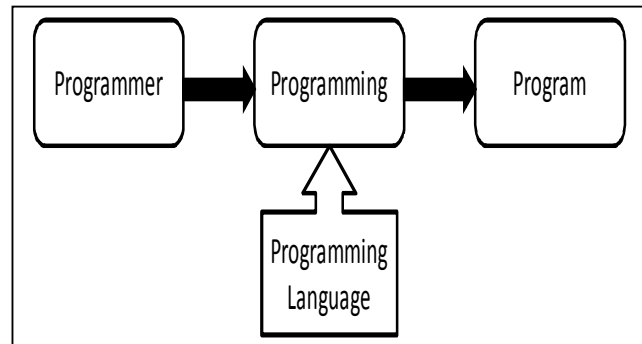


Figure 3: Programming Concepts

Teaching programming is not an easy task, so we emphasize that students are going to learn a new language, which means they need to be familiar with the grammars “syntax” behind the language in order to formulate a clear understandable coding statement as well as the alphabets. In any new language in our world there are two solid fundamentals, the alphabet of the language and the grammar or structure. These two fundamentals are necessary to write and speak well understandable language statements and communicate with the environment around us. The same happens when we teach programming language; we make sure that students are aware that they will learn a new language in which they can communicate with the computer easier.

The second step in our experience is to familiarize students with the different problems that must be solved through software, explain to them how to read the problem carefully and analyze it, then design the solution out. The step by step problem solving technique is called the Algorithm writing. We encourage and teach students how to write Algorithms before coding, because as teachers we noticed that students are facing difficulties in understanding the question to be programmed and the problem to be solved, reading the problem at least twice, determine the requirements from the problem, analyze the input, output and the formula, then design the solution in an English step by step sequence, after that trace it by hand, and later translate each algorithm statement or step written into a code statement following the correct syntax. The algorithm writing procedure made an improvement with students on how to read the problem, understand it and solve it, especially when the students read the problem well, and know how to read it, then analyze and determine what input should be entered to the program, what output should the program display at the end, and if there is

any mathematical formula that must be used in solving the problem; after that start designing and writing out the logical algorithmical steps to solve out the problem before translating it into programming statements which are made up from the syntax.

Our third step in teaching programming to engineering students is to link it with the real life by adding more problems that are related to their area of specialization, rather than discussing business related problems or mathematical problems. We supply engineering and scientific problems to students. By this we noticed that students are more interested in programming because they realize that they are able to do such real applications to which they deal with in their daily life and their area of specialization. In addition we try to link what the students are taking in other courses like physics for example and try to ask them to convert the theoretical problem into a java program to solve it. By these students feel confident that they can implement programs that may help them solving other courses they are studying.

The fourth experience, was to teach them how to program some simple games, like drawing face game, tic-tack toe game, paper, scissor and rock game, and others, by this; students experience that programming course becomes more enjoyable than only a classic course full of theory.

The fifth experience implemented was to establish a Tutorial session about 1 hour, 30 minutes per week. This session was implemented to enable students just to solve extra challenging questions. It was noticed, that students' performance in the course increased when they started attending the Tutorial sessions and experience the solution of different programming problems that widen their knowledge of thinking and adds more programming experience to them. The Tutorial sessions were given by a senior Software Engineering student, and the aim behind this, is to make the sessions more close and conducted in a student friendly environment.

Our sixth experience which we conducted, is implementing such funny competitions in programming, especially during making programming quizzes, we initiate the idea of the "*Programmer of the Week*", this is a certificate we issue in addition to a small gift for the excellent programmer student every week. This idea added more value to love programming and increase the level of studying in order to be the Programmer of the Week, by this students were motivated to study and participate in solving the programming problem and had spirit of challenge to achieve "*the programmer of the week*".

The last experience is to distribute small gifts of candies to students who ask more questions in programming classes or those who answer Instructor's questions or participate. This idea will add value to other students to pay attention inside class and make programming classes more fun and enjoyable.

Teaching programming is a world full of experience and challenge. It is believed that innovation is part of teaching and learning programming courses. It is advisable to programming teachers to be patient with student, motivate and support students. On the other hand students are advised to regularly attend programming classes and always practice as much as

they can the programs, play with them, and understand the results; this is the only way to be a good programmer in the future.

VIII. CONCLUSION AND FUTURE WORK

Programming is a world where programmers communicate with computers using a programming language. It is an interesting world to live in, but there are a lot of difficulties that may be faced. Teaching programming and studying programming is a journey full of challenge and motivation. For students it was difficult to learn a new language in a short period of time. This language will then be used to deal with computers.

Teaching programming put a lot of efforts on teachers to succeed in delivering the course in the right way. Many ideas and methodologies were implemented in this paper that add value in teaching programming, by putting a good plan, focusing on laboratory sessions, implement real life application, and much more. After this experience, student performance, experience, and strength in writing programs increased. A comparison study was made at the end of each academic year for the past 2 years (2010 – 2012) when we started implementing these teaching methodologies shows that student performance and grades raised up more than 70% in comparison to the previous years. In addition students started to feel comfortable and happy when they study the course.

On the other hand, another idea in the way of making programming languages easier to learn and use, is trying to make these languages readable, so users will avoid typing and start programming by voice. Several studies and researches have been done in the field of programming by voice. The big improvement in speech recognition software encourages the researchers to make use of them to reach the goal of programming by voice. [13]

A project idea under a research study at ALHOSN University titled "Programming by Speech Application" is to be implemented. The idea focuses mainly on beginners in programming. Taking into consideration the importance of writing algorithms as a step in the programming process, the students are asked to enter, by speech, the algorithm of the program to be developed in pseudo code format using predefined natural English keywords. After recognizing the speech, the system runs a validation process to validate the syntax of the entered pseudo code, and provide correction options if necessary. If the pseudo code is error free, the system will generate the corresponding code using selected programming language (currently java). The student can view and match between the entered pseudo code and the generated code in an easy way that help him/her to learn the code native syntax easily and fast.

Through the experience, steps and the teaching efforts that were covered in the paper, learning programming has changed the classical and dramatic way of teaching and learning into a real noticed piece of cake course. New ideas of teaching can be easily generated by experience and programming instructors are encouraged to love what they teach in order for students to love what programming they learn.

REFERENCES

- [1] Cooper, S., Dann, W., & Pausch, R. 2000. Alice: A 3-D tool for introductory programming concepts. In Proceedings of the 5th Annual CCSC Northeastern Conference 2000. Ramapo, NJ, April, 28-29, 2000.
- [2] Trevor Nesbit, The Teaching of Introductory Programming: Issues of Context, This quality assured paper appeared at the 22nd Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2009), Napier, New Zealand. Samuel Mann and Michael Verhaart (Eds). Reproduction for academic, not-for profit purposes permitted provided this text is included. www.naccq.ac.nz
- [3] Hu, M. (2003): A Case Study in Teaching Adult Students Computer Programming. In Proceedings of the 16th Annual NACCQ, Palmerston North, New Zealand, pp. 287-290
- [4] D. Budny, L. Lund, J. Viperman, J.L.I.I. Patzer, "Four steps to teaching C programming," *Frontiers in Education*, Annual, vol. 3, pp. F1G18-22, 32nd Annual Frontiers in Education (FIE'02), 2002.
- [5] Teaching Programming Principles through a Game Engine, Cristiane Camilo Hernandez, Luciano Silva, Rafael Alencar Segura, Juliano Schimiguel, Manuel Fernández, Paradel Ledón, et al. in *SciencesNew York* (2010)
- [6] On the Development of a Programming Teaching Tool: The Effect of Teaching by Templates on the Learning Process. Authors: Al-Imamy, Samer; Alizadeh, Javanshir; Nour, Mohamed A. *Journal of Information Technology Education*, v5 p271-283 2006
- [7] Matthiasdottir, A. 2006. How to Teach Programming Languages to Novice Students? Lecturing or Not? In Proceedings of the International Conference on Computer Systems and Technologies. (Bulgaria, June 15-16, 2006) 13, 1--6.
- [8] Expert tutoring system for teaching computer programming languages, M.M. El-Khouly*, B.H. Far, Z. Koono. *Expert Systems with Applications* 18 (2000) 27-32
- [9] A Study of the Difficulties of Novice Programmers. Essi Lahtinen, Kirsti AlaMutka, HannuMatti-Järvinen
- [10] M. A. Perez-Quinones, Teaching history of programming languages to undergraduate students, Proceedings of the 28th Annual Frontiers in Education, p.301-306, November 04-07, 1998
- [11] "Top 10 Best Programming Languages 2012," OMGTop Tens.com, 16 July 2012. [Online]. Available: <http://omgtopens.com/misc/education/top-10-best-programming-languages-2012/>. [Accessed 27 October 2012].
- [12] "JCreator," [Online]. Available: www.JCreator.com. [Accessed October 2012].
- [13] Snell, L. An Investigation Into Programming by Voice and Development of a toolkit for Writing Voce-Controlled Applications. M Eng. Report. Imperial college of Science, Technology and Medicine, London. June, 2000.