

Applying and Improving Multidisciplinary Teaching Techniques to the Programming Classroom Environment

LindaLee Massoud JD¹, Steve Hallman DBA², Michel Plaisent Ph.D.³ and Prosper Bernard, PhD⁴

¹ IT Department, Computer Information Studies Program Mott Community College, Flint, MI USA,

²Higher Education and Business Consultant, Accreditation Mentor / Evaluator,

^{3,4}Dept. Management and Technology, University of Quebec in Montreal, Canada

Abstract: As we move into the 21st century, there will be a significant increase in the need for computer coders and programmers in STEM and other technical jobs. The broadening of the student base, and their respective variations of skills, necessitates a review and reconsideration of the way computer programming is taught if it is to attain maximum success. This paper approaches computer programming education from a multidisciplinary perspective, comparing it to the learning of a language as a foreign language versus a second language. Various techniques are evaluated within the parameters of both Dale's Cone of Experience and Bloom's Taxonomy to craft an effective learning environment.

Keywords: computer programming, coding, programming, teaching techniques, Dale's Cone of Experience, Bloom's Taxonomy, education, educating

1. Introduction

There are two approaches to learning spoken languages, such as Spanish or Arabic: either as a foreign language or as a second language. The distinction between the two is the eventual use for the language. Differentiated instructional techniques can enhance learning success. By comparison, in the field of computer programming, one might distinguish between coders and programmers. The meanings of these two terms are evolving (Prottzman, 2015). For the purposes of this paper, coders learn the terms and concepts of programming to achieve certain goals (such as engineers); programmers specialize in formal programming as a profession. Differentiated instructional techniques can enhance learning success for both groups. Applying language learning techniques to the computer programming classroom, particularly with reference to Dale's Cone of Experience and Bloom's Taxonomy, will result in more successful and competent coders and programmers.

2. Foreign Language Goals and Approaches

The foreign language (FL) approach is used in situations where one is learning a language for academic or scientific purpose rather than for daily use. Instruction is classroom or book/web based, and the focus is on vocabulary, grammar, and broad subject matter capabilities (although formal, probably stilted and accented). The final goal of the typical foreign language student is to gain exposure to another language and culture, to be able to read technical materials or conduct limited spoken communications on limited topics, or simply to enhance critical thinking skills (foreign vs second language learning, n.d.).

3. Second Language Goals and Approaches

Distinguished from the foreign language approach is the second language (SL) approach to learning a language. Students of this method intend to use the language daily and for personal interactions. Rather than being classroom/book-based, this method emphasizes real-world exposures and communication events. Second

language learners may have a narrower range of vocabulary skills, and may have residual grammar errors, but their inter-personal communication skills exceed those of the foreign language learner. The final goal of the second language learner is to conduct practical, daily life communications in a variety of real-world situations. (foreign vs second language learning, n.d.)

As a recent, real-world example, this author was speaking with a student who proudly stated that he had taken two semesters of [a certain language] class. Upon giving him the standard morning greeting in that language, he looked shocked. He was not able to give the standard reply, either then or later. Clearly this student had taken two semesters of a foreign language, but had no practical second language training in order to even say “Good morning” back to the greeter!

4. FL vs. SL Classroom Approaches

Continuing the distinction between the two methods, consider the classrooms in which students gain the FL or SL skills. Foreign language learners would typically take a college class that emphasizes formal reading and writing skills with some supplemental emphasis on speaking and listening. (Some of these instructional techniques are evolving as students demand more practical application exposure.) Second language classrooms would be very much scenario and conversation-based. Students learn to find ways to communicate successfully with their limited set of vocabulary and grammar. Achieving successful communication is more important than using the language “correctly.” Community education classes, elementary school classes, and immigrant classes are examples of this learning approach.

5. FL-SL Concepts Applied to Computer Programming Environment

Although it might not seem like it to programmers, novice programming students are in a situation not unlike foreign/second language students. The terms, concepts, grammar, and procedures are all completely foreign to neophyte programmers/coders. Similarly, students may be interested in learning programming as a “foreign language” or as a “second language.” Specialized teaching techniques make the learning of another language more comprehensible, whether conducted in the first language or second language. Relevant examples of these techniques include linking new knowledge to prior knowledge, directly teaching relevant vocabulary and concepts, and using cooperative learning strategies (Haynes, 2010).

Foreign language programming might have a final goal of engineer, networking, or other technical field that is not programming per se but necessitates the ability to read, understand, interpret, and communicate about computer programs. The focus for this group is on vocabulary and broad concepts rather than specific details and techniques. Whereas a second language technique might incorporate more of the “functional/communicative” orientation to learning methods, a foreign language approach might be more akin to the “structural/linguistic” approach in which individual elements of the language are taught intentionally and much meta-cognitive content (learning about the language) is also included (Mora, n.d.)

6. Dale’s Cone of Experience Applied to Computer Programming

It is well known that the more interactive and immersive the learning environment, the better the comprehension and the longer the retention. (Benjes-Small, 2014). In a typical “old-school” programming classroom, the professor would demonstrate all or parts of several programs as a means of teaching the theory. When this author first took some programming languages, we only had the professor’s lecture and demonstration of a single program. There was no textbook. Geho and McDowell (2015) suggest that learning through experimenting and participating is an effective approach to academic success, as it contributes to the emergence of creative solutions, improving critical thinking capacity, and communication skills. Of course, the 21st century is a different world, and the professor can afford to design much more of a flipped classroom approach.

Flipped learning is a pedagogical approach in which direct instruction moves from the group learning space to the individual learning space, and the resulting group space is transformed into a dynamic, interactive learning

environment where the educator guides students as they apply concepts and engage creatively in the subject matter (Flipped Learning Network, 2014).

According to Hoffman (2014), there is an urgent need to replace single instructional tasks that are only assignment goals to promote learning environments facilitating knowledge application and deeper understanding. According to Mazur (2013), a flipped classroom is a new approach that respects several theoretical constructs:

- The constructivist pedagogy, is a dimension which focuses on the knowledge and consequence of the learner internalization of the new concepts and information following an integration effort thru thinking;
- Inquiry defers to the intimate intrinsic curiosity of the learner that searches for a view to better-understand. This is accomplish by using more or less formal investigation methods;
- Student-centered learning inverts the role of teachers who act more as planners and animators of students who regulate their own learning pace and methods. Active learning is the expected impact produced by diverse stimulations produced by peer communication, video, documents and simulation, and commonly name as virtual reality.

Certainly, having the professor talk, write, and produce a computer program in front of the students (via a projected computer image) is valuable for a 50% retention level. There is no doubt of the value of this, particularly in the beginning. An even higher level of retention can be achieved, particularly in the first few weeks, by having the students write the program simultaneously with the instructor. Adding the “doing” component would bring the retention rate higher (even though there is no direct listing on the “cone” pyramid). Alternatively, there are many, many videos available online to demonstrate skills at the see-hear-watch level.

A more effective use of class time is the flipped classroom approach where activities at the lower levels would be accomplished outside of class. Also, while some students find the standard textbook valuable, and will use the meta-cognitive approach of personally (“grinding”) working through examples (McGuire), most 21st century students skip the “read” level entirely.

Bishop & Verleger (2013) conducted an extensive review of the literature on flipped classroom that led them to conclude that reverse classes are generally perceived positively, but all too often these studies suffer from a lack of validity due to the fact that they are perceptual studies, based on a single class at a given semester. Hence, this lack of rigor is lamented by Goodwin and Miller (2013), who advocate controlled experimental research on this subject, to more rigorously evaluate their real effectiveness.

An interesting new instructional possibility is merging the two techniques into a focused approach using “information and communication technology” (ICT) methods in the computer classroom, as is being investigated for the military classroom where a range of both knowledge, culture and practical language skill is required. The method incorporates intensive use of online and computer-based instructional method for certain skills, allowing the face-to-face time with the instructor to focus on didactic activities designed to build on previous skills. (Niculescu, 2017)

With basic knowledge and understanding level activities taken care of outside the classroom, actual class time can be devoted to activities at the 70% SAY and 90% SAY and DO levels. The SAY level might be difficult to implement in a larger class unless they are divided into small groups to “present” their novel program. (A final project might be a justifiable exception to the SAY rule. Consider, though, that at 10 minutes per student, 5 minutes per presentation plus transition time between projects -- even a small, standard classroom size would require a couple of class periods to present all projects. And the other students are guaranteed to claim boredom. Ways around this include dividing the group into smaller presentation sets, in alternating class periods, and requiring a detailed and thoughtful analysis by listening students.)

An alternative “SAY-level” learning activity would be to require substantial documentation of a program. Documentation is a highly-sought habit by employers. “No job is finished until the paperwork is done” is a laudable theoretical goal, but forcing the early development of a habit is a more effective approach than mere platitudes about its value. The professor should require a variety of documentation features, which might vary

from assignment to assignment, so students have exposure to the range of commenting methods that might be required in various companies. It would be valuable, (but maybe not realistic due to time limitations), to include exposure to version control software. There may not be time in the curriculum to use actual version-control software. Revision control is an excellent way to combat the problem of sharing files between workers.

The best retention in Dale's Cone of Experience comes from the SAY and DO level of learning. Multi-sensory learning is always the most effective learning method, and SAY/DO exemplifies the value of this approach. Students might take on roles of teacher and student, or supervisor and employee, rotating in subsequent assignments. Even brief exercises using these roles help students with both retention of the material and preparation for the work world.

Teamwork and presentations require significant extra time and effort on the part of both professor and students, and both are likely to complain, but they are by far the most effective, practical, and long-lasting method of learning, and the best to prepare students for the work world, whether as a FL or SL type of user. Efforts toward helping students approach their work from a “growth” mindset rather than “fixed” mindset will also enhance their work preparation (Dreon, 2017). Crafting effectual assignments that fit within the limited framework of a class period require thinking and planning, but can be accomplished.

Finally, intrinsic motivation is a key-factor to learning, along with self-determination and appropriate social interactions (Thorsell, 2014).

7. Bloom’s Taxonomy Applied in FL-SL

Having established that courses could be taught with a different emphasis depending on the intended final goal, reviewed the range of sensory experiences that promote the best learning, the next task is to consider what kinds of exercises would accomplish the goal. Bloom’s taxonomy is a useful vehicle for such an analysis.

Bloom’s taxonomy lists levels that represent “depth” of learning, from more surface levels to more deep levels. They are listed in progressive order below, with columns for both FL and SL approaches, and sample activities at each level (Aromstrong, n.d.).

TABLE I. Bloom’s Taxonomy Levels to Foreign and Second Language Programming Examples

Bloom’s Level	Foreign Language	Second Language
Remember Recall facts and basic concepts	Identify vocabulary, grammar, theory	Identify vocabulary, grammar, theory
Understand Explain ideas and concepts	Explain the purpose and use of certain coding structures Explain the function and result of a set of code	Recognize similarities between languages Explain the function and result of a set of code
Apply Use info in new situations	Given one/several sample programs, create or modify a novel (but similar) program demonstrating those skills	Given one/several sample programs, apply the programming concepts to create a new and different program
Analyze Draw connections	Debug the error in an existing program (may or may not include repair)	Debug and repair the error in an existing program Contrast syntaxes of various languages
Evaluate Justify a stand or decision	Might not be used	Given a certain algorithm, offer and evaluate the feasibility of an alternative coding approach. (Example: rewrite a program portion using a different loop type)
Create Produce original work	Might not be used	Given a set of “constraints,” produce a novel program that accomplishes a task while simultaneously demonstrating the required concepts. (This would be a final project.)

8. Applying Bloom's Taxonomy in a Blended Focus Classroom

The optimal instructional environment would have separate classes for FL and SL learners. The instructional emphasis, methods, and assignments would be different. This might be possible in a large school with multiple sections of the same course or a targeted training program (such as an engineering program). The reality, however, is that most classrooms will consist of both kinds of students, and the instructor is responsible for creating a balance between instructional methods.

According to meta-cognition theory, the most effective way to help students in a blended environment achieve the maximum academic success is for them to understand the difference in FL vs SL goals and to learn how to recognize and apply Bloom's taxonomy to their learning experience. In short, McGuire supported to help the students become educated learners who accept responsibility for their own learning.

The time spent in class helping develop these techniques provides a high dividend pay-off, but the professor will have to help the students realize the value of the meta-cognition skill. Some authors recommend allowing students to perform poorly on the first test, while using their existing study skills, as a way to add "incentive" and willingness to learn the new thinking approaches. This author suggests an alternative with less consequence to the final grade: provide a "low stakes quiz" prior to the first test to enable them to realize their need for new techniques and implement them prior to the first high-stakes grade.

As for content in a blended classroom, careful attention to Bloom would result in a variety of exercises, at multiple levels, designed for both groups. In a 1st (or 2nd) semester class, where the primary goal is to learn foundational programming concepts, the goals for both FL and SL learners could be similar. Second-year courses would probably focus on skills for future programmers, using a SL approach.

Outcome-based learning is the current focus of all levels of educational programming. Well written course objectives should guide the development of all assignments and assessments. Course objectives written across all levels of Bloom's taxonomy help ensure the best educational result.

9. Assessment Ideas for Each Bloom's Level

While there are many possible ways to implement programming assignments, a few sample suggestions may trigger each instructor to have more ideas based on individual past experience.

9.1. Remember and Understand

These levels of knowledge are most effectively assessed using questions in a Learning Management System (LMS). . Typical textbooks included knowledge and some understanding level questions that can be imported into the LMS. Informally, though, the professor should always make an effort to incorporate interactive Q&A into classroom discussions. Frequently, students know the "buzz words" but do not actually understand the foundational concept behind the word. Interactive conversation would bring these gaps to light.

9.2. Apply and Analyze

The application and analysis levels of Bloom's taxonomy help intermediate-level students develop the ability to apply knowledge to specific circumstances. These activities are preparatory to later evaluate and create level activities.

- **Objective-style questions.** If carefully written, activities at these levels are possible in an auto-graded format and could be used as a supplement to hands-on activities to ensure coverage of the course objectives.

o Sample question: You will receive a list of numbers (of known length) from an external source and manipulate or evaluate those numbers in your program. Which kind of loop would be most effective to use in this situation? (a) Count-controlled loop (b) While loop (c) Do-while loop

- **Instructor-led demonstrations.** Help students learn the thinking and logic process that they will apply later. At this level, "more and shorter" assignments are appropriate because "practice makes perfect." Significant instructor input is the most valuable at this level of skill development.

- o Short programs that demonstrate a single concept at a time. Example: In a function, get a number from the user. Send that number to a different function to be used in a calculation and returned to the main program.

- o Small enough that students can type along. Note, however, that some students learn better by watching only, and they should be accommodated by providing the code on completion. (Many other students may have run into syntax errors by typing and would also appreciate receiving the working code.) Add a small follow-up assignment using that code to ensure that they actually interacted with it.

- o Instructor talk-and-code format so students will learn proper thinking habits.

- **Program talk-through.** Easily suitable for on-demand video, another apply/analyze activity would be the program code “talk-through,” without simultaneous code writing. Show the code on the screen, in large enough font size to read easily, and talk through the code logic without actually writing the code.

- **Student coding activities.** Given a basic program created by instructor or another student, the “next” student enhances the program as specified. (This exercise simulates the work experience of entering a new position after a previous employee leaves.) This exercise could even be continued to a second level. Students will also see the compelling value of in-depth documentation of logic and thought processes.

- o Example: Original program uses a while loop. Student converts the program to a do-while loop.

- o Example: Original program is self-contained in a main function. Student converts the program to a “functionalized” version.

Understandably, this approach to teaching initially requires a significant effort by the instructor to develop the portfolio of applicable assignments. Most books teach the skills in a similar order, though, and updates are minimally different, so a careful development once can be reused for numerous years. Furthermore, the assignments do not have to be tied to a particular textbook as long as the course content is duplicated for all sections and modalities of delivery. This is an area where a network of colleagues could have a shared materials library for all to use/modify.

Allowing some, individual creativity in the development of specific exercises, based on the individual professor’s background and experience, can enhance the satisfaction with the classroom experience – as long as the activities track the course and project objectives. At the same time, using consistent assessment tools and measurements can help ensure consistency of learning outcomes for students, faculty, and other stakeholders (the board, committees, councils, and hiring entities).

9.3. Evaluate

For mid-level assignments assessing application, evaluation, and creation skills, more specialized assignment options might include:

- One assignment of each type (in early lessons) or alternating between chapters;
- Option to choose one assignment or the other (both assignments would need to be at similar difficulty levels);
- Rotating roles in high-level simulation activities.
- Student creates a program, then prepared a “work directive” (set of instructions) for another student to re-create the program. In addition to the actual experience of writing the program, both side of this exercise require significant evaluation activities.

- o The creator of the program must document the request in such a way that a programmer can understand and implement it independently.

- o The student writing the program must figure out what the instructions mean and determine the best way to implement them. (And perhaps student #2 will learn how to go back and ask for clarification, which is a work-based skill itself.)

- o The most valuable part of this activity is the post exercise review, where both parties evaluate the clarity of the instructions and the accuracy of the implementation of them.
- Student(s) write a certain program. Another group evaluates the logic, code, and documentation, and provides feedback to the programmer.

9.4. Create

Create level tasks are the most fun for everyone, but they also take the most time. Due to time limitations, use them as a final semester project rather than an individual program. In addition to the book, some sources of creative topics include:

- Instructor's or students' personal experience or interest.
- Group/class determines a scenario and divides the major topic into individual parts to be merged together at the end.
- Given a generalized statement of goal, allow multiple teams to develop simultaneous projects, then compare/contrast at the end.

10. Value of Team Projects

Of the most benefit to the students, however, are exercises at the Create level of Bloom's Taxonomy and that use team work. An ideal scenario that meets the needs of both FL and SL groups would be teamwork programs with heterogeneous groups that would require role-play from various team members, all appropriately documented. Although this approach would be particularly useful for a final project assignment, this author starts students on team projects as soon as they learn functions, early in the semester. Several similar, practical exercises are offered by Lathrop in the Recommendations section (Lathrop, 2010).

The most valuable lesson students learn from this exercise is the need to be exceedingly precise in following directions (function prototypes, variable declarations, variable naming, and return methods). It takes numerous, repeated attempts for the students to learn this level of detail, but after that they can produce much more extensive programs during a single class period. In addition, students quickly learn the process of checking and confirming details with each other (teamwork). It is both gratifying and valuable to see them checking and debugging with other team members. Since team programming is how the “real world” works, all attempts at working with team work at the Create level of Bloom's prepare students most effectively for both FL and SL uses of programming. (Students later report that teamwork, and being forced to look up details to accomplish program tasks, were the two most valuable skills they carried to the market.)

Even as of 2015, jobs that require coding skills were growing at a rate 12% higher than the market average. “Computer programming jobs may be declining, but coding is becoming the most in-demand skill across industries” (Dishman, 2016)

Computer programming, or coding, is a skill that will be in increasing demand in the next few years (Maxwell, 2016). While there are numerous occupations that currently require coding, over 50% of the STEM jobs by 2020 are projected to be in a computer-science field (FACTSHEET: New Commitments to Support Computer Science Education, 2014). Many related jobs will need some level of coding skill. As of 2016, over 250 million children been introduced to coding starting at very young ages (Vickery, 2016); they will be filtering into our colleges in the next few years. As the incoming skill level builds, the way programming classes are taught will likewise need to be modified. Both “foreign language” and “second language” approaches are appropriate instructional methods for certain occupational purposes. The more multi-sensory the learning approaches, the better the retention of the material. Requiring assignments at all levels of Bloom's taxonomy, with emphasis on teamwork and exploration/sharing, will result in the best learning outcome to prepare our students for the high tech jobs of the 21st century. Additional information on this subject is available in the Occupational Outlook Handbook (<https://www.bls.gov/ooh/>).

11. Conclusion

We are moving toward a more global society, with expected delivery of consistent training outcomes. Colleges and universities are becoming more accountable to external stakeholders. The mission and objectives that start at the top trickle down to the individual professor in the section of the course. Educators and stakeholders believe that students should possess strong content mastery, as well as critical thinking, communication, collaboration, and creativity skills. To achieve these goals, instructional methodologies must become more standardized and consistent. Instructional approaches that combine proven educational theory with solid outcome-based course objectives will lead to the most efficacious educational result.

12. References

- [1] Aromstrong, P. (n.d.). *Bloom's Taxonomy*. Retrieved from Center for Teaching: <https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/>
- [2] Benjes-Small, C. (2014, Jan 13). *Tales of the Undead . . . Learning Theories: The Learning Pyramid*. Retrieved from ACRLlog: <http://acrllog.org/tag/learning-theories/>
- [3] Dishman, L. (2016, June 14). *Why Coding Is Still the Most Important Job Skill of the Future*. Retrieved from Fast Company: <https://www.fastcompany.com/3060883/why-coding-is-the-job-skill-of-the-future-for-everyone>
- [4] Dreon, O. (2017, August-Sept). *Taking Collaboration Seriously*. Retrieved from The Teaching Professor: <https://www.magnapubs.com/newsletter/the-teaching-professor/137/Taking-Collaboration-Seriously-14581-1.html>
- [5] FACTSHEET: *New Commitments to Support Computer Science Education*. (2014, December 08). Retrieved from The White House: <https://obamawhitehouse.archives.gov/the-press-office/2014/12/08/fact-sheet-new-commitments-support-computer-science-education>
- [6] *foreign vs second language learning*. (n.d.). Retrieved from Blackwell Reference Online: http://www.blackwellreference.com/public/tocnode?id=g9780631214823_chunk_g978063121482310_ss1-6
- [7] Haynes, J. (2010). *Seven Teaching Strategies for Classroom Teachers of ELLs*. Retrieved from everythingESL.net: http://www.everythingesl.net/inservices/seven_teaching_strategies_clas_06140.php
- [8] Lathrop, S. (2010). *Teaching Techniques for Advanced Computer Programming*. Retrieved from http://www.usma.edu/cfe/Literature/Lathrop_10.pdf
- [9] Maxwell, D. (2016, July 4). *Coding is a Necessary Part of 21st Century Education*. Retrieved from The Educator: <https://www.theeducator.com/blog/coding-necessary-part-21st-century-education/>
- [10] McGuire, S. (n.d.). Teach Students How to Learn. Retrieved July 20, 2017
- [11] Mora, J. K. (n.d.). *Second and Foreign Language Teaching Methods*. Retrieved from MoraModules: <http://moramodules.com/ALMMMethods.htm>
- [12] Niculescu, B.-O. &. (2017). Exploiting Information and Communication Technologies in Teaching Foreign Language. *Buletin Stiintific*, 38-45. doi: doi:10.1515/bsaft-2017-0006
<https://doi.org/10.1515/bsaft-2017-0006>
- [13] Prottzman, K. (2015, April 12). *Coding vs. Programming -- Battle of the Term!* Retrieved from Huffington Post: http://www.huffingtonpost.com/kiki-prottzman/coding-vs-programming-bat_b_7042816.html
- [14] Vickery, E. (2016, August 23). *Coding and Computer Science: Necessary Courses in 21st Century Schools*. Retrieved from Partnership for 21st Century Learning: <http://www.p21.org/news-events/p21blog/1984-coding-and-computer-science-necessary-courses-in-21st-century-schools>.