

Learning by Interactive Programming (L.I.P.)

Emiliano C. De Laurentiis

This article is about the value of programming as a teaching tool - not to teach programming, but rather to learn other subjects. That is, the use of programming as a medium that students use to illustrate their understanding of a concept. For example, if a student can program a computer to draw circles using basic trigonometric functions then the student understands the concept of circle at a very fundamental level. If a student develops a simulation that models the economic functions of a city, then that student has learned some important lessons about such an economy. I call this use of programming for teaching and learning, Learning by Interactive Programming, or L.I.P.

It is interactive because whether the program works or not is immediate feedback of the student's understanding of the problem. Once they have learned the language of programming, the main stumbling block in writing a program is their understanding of the concepts related to the problem at hand. They are learning by the research that they conduct to solve their programming problems. For example, to draw a circle they will have to ask themselves such fundamental questions as "what is the relationship between a point and a circle?", "what is the center of a circle?", "what is the relationship between the angle from the center to a point on the circumference and a point adjacent to that point?" etc.

L.I.P. can be used to benefit any curriculum. To program a computer to draw a circle, for example, you need to understand circles. In order to program a computer to check for certain types of grammatical errors, you need to understand and recognize the type of grammatical errors that are typically made. The process of programming forces a student to understand the knowledge domain at a deeper level. The proof of their understanding is whether the program works.

L.I.P.'s greatest strength may be that it is highly motivating. Once students have the basics of programming, and as long as the tasks assigned to them are within their skill level, they become highly immersed in the programming task. The combination of immediate feedback and satisfaction of seeing the program start to build, as each programming component is completed, and problems are solved, is very engaging.

Non-programmers usually see the learning of a programming language as the greatest challenge in L.I.P. But today there are a multitude of computer languages, at varying levels of difficulty, that make it pretty easy to start programming. There are computer languages that focus on language manipulation (chat -bots, for instance), there are languages that focus on geometry (such as LOGO), and others that are used to create role playing games. The specialized programming languages are easier to learn, but the generalized languages give students the greatest power. Selecting the computer language or languages to work with should be based on curricular goals (it might be to program musical concepts, for example), on the skills and commitment of the faculty, and the experience of the students. For example students in elementary school may learn to use LOGO but in middle school may progress to Java.

Every student should, by the time they enter middle school, have a basic foundation of programming concepts so that they can jump from one computer language to another with relative ease. Personally, I have taught myself six computer languages. I can learn a new one - at least to accomplish simple tasks, in as little as a day. It is like people who learn many human languages.

Once you have learned your second language you can learn a third and a fourth with progressive ease (assuming you stay within a linguistic base such as romance languages).

It should be the goal of every school district to teach programming at a very early age, and then to use that skill in every subject as often as possible. Teachers may never catch up with students' abilities, but that is how it should be. The teachers' role is to set goals and to evaluate how well the students have accomplished their projects - which is plainly evident by a functioning computer program. Consultants and industry volunteers can help with the technical details of programming. Such a valuable teaching tool does need one important ingredient from schools, though, and that is leadership!

Feel free to reprint, but please provide a link to www.knowledgeenvironments.com with the line: "Copyright 2006 by Emiliano De Laurentiis" To learn more call 413-241-7314 or e-mail edl@knowledgeenvironments.com