

[Learning Apps](#) | [Features](#)

How to Create Your Own Learning 'App'

Simple (and hard) ways for students and educators to start developing learning apps

By Kim Fortson • 08/08/12

Web guru Anthony Luscre believes "if you're not teaching over your head, you're not doing an effective job." He applies this principle particularly when encouraging teachers to develop and implement apps in their classrooms. Just how far above one's head, however, depends on how involved a teacher wants to get. Incorporating apps into the classroom is a process that can take an hour or six months, depending on what instructors wish to achieve.

Luscre has spent the last 16 years as director of technology for [Mogadore Local Schools](#) in Mogadore, OH, where he coordinates everything from the facilitation of the district's website to the integration of technology through professional development. A frequent conference speaker, Luscre discusses strategies for building and using apps in the classroom in one of his talks, entitled "Got a Problem? Create an App for That."

Over the course of 60 minutes Luscre highlights a variety of strategies that comprise app development, but first, he said, you might have to rethink your definition of an app.

"[I use] the term 'app' to describe any type of process we might be doing," he explained. "Something as simple as an app that would allow students to subtract numbers that require borrowing."

Adopting this definition as a foundation for his seminar, Luscre takes his audience through a process for creating what he calls "pseudo apps"—approximations or rough sketches that contain the information needed to build the apps, if not the actual software. At the end of the seminar, he establishes a basic knowledge of the steps necessary to take pseudo apps from concept to reality.

"The idea of the class is really to approach apps in a couple of different ways," Luscre said.

To do so, Luscre breaks down the process of app development into a series of steps: flowcharting, functional flowcharting, building real apps, and actually coding them.

"I teach critical thinking," Luscre said. "It's not so much the end product of the app as much as teaching students to think in that type of way."

He begins with flowcharting, or transcribing what an app needs to accomplish—its input and output—which he does by diagramming software with pen and paper. In addition to being a way for a teacher to visualize the app he or she wants to create, flowcharting can be used as an activity in the classroom.

"Flowcharts are the most 'bang for the buck.' [They are] very easy to start using and very powerful for visual learners," Luscre said.

Next, Luscre focuses on what he calls functional flowcharts, the most basic of which are spreadsheets. Spreadsheets actually generate an output when they receive an input, thereby achieving a flowchart's utility. Luscre's goal is to stretch the typical function of a spreadsheet, noting that his teachers have used them for generating median, mode, and average; prescribing values to coins; and even producing the names of characters in *Romeo and Juliet*.

"Spreadsheets work as an algorithm. You're building an algorithm for doing whatever you want to do," Luscre said. He cited Microsoft's Visio Software as another program that can create functional flowcharts.

For teachers who want to use true apps in their classroom, Luscre discusses ways that do not necessarily require coding. Websites like [Apps Geyser](#), for example, allow users to build their apps for free simply by copying and pasting existing code and content.

The final step is learning how to code one's own apps, a skill, Luscre says, that "isn't for everyone," though he believes the benefits that come from learning the steps behind making apps are universal.

"[Using apps in the classroom] can more effectively teach students who are visual and kinesthetic learners," Luscre said. "Most teachers tend to favor the linguistic side of things...but there are students whose main skill might not be reading or writing, but manipulating visual objects. Teachers can actually go through and teach some of their lessons in the same way a program would work--not depending on language but depending on visuals to have someone work through the process."

Luscre offers a wealth of information, websites, and examples for implementing technology in the classroom on his own website, [SearchFindKnow](#), which provides supplementary materials and in-depth information on his lectures.

From Teacher to Coder

Teachers who want to reach far enough above their heads to learn how to code are signing up for a long, tedious process--one that can take up to six months. But New York science teacher Fred Feraco believes it's worth it. Feraco learned the basics of multiple coding languages in order to program his educational "[Buddy](#)" apps, a series of iOS apps [designed to help his students study](#). Most are specific to the New York Regents exams.

Feraco's journey began at the beginning of the last school year, when he decided he wanted to learn how to build his own apps and paid to be a part of [Apple's Developer program](#), where he then had access to the iOS coding software XCode. He began fidgeting on weekends with the program, which uses the C Command coding language.

"At first it's very difficult," Feraco said. "I must have sat around for a month just playing with it."

Feraco and a friend decided to sign up for an Introduction to Java Programming course at Columbia University to familiarize themselves with multiple programming languages and to supplement their self-learning.

The program was difficult, and took most of Feraco's spare time. "Computer science sounds really fun and cool, but there's a lot of math involved, there's a lot of discipline involved that you need to use to code and develop these apps to get them approved. There's a lot of patience," Feraco said.

What comes with these skills, however, is the satisfaction of seeing one's own apps in the iTunes store and the ability to troubleshoot when there are glitches.

Though Feraco learned Java programming in his class, which is used for Android apps, he only codes them for iOS--but he believes the skills he gained with that knowledge have helped him in XCode.

"There are things that need to be very particular no matter what type of programming you use," he said. "So just learning the habits and learning how to write different parts of the code, and how particular it needs to be, and the history of it, gave me a good background before I jumped right in."

Even then, Feraco waited months for his apps to be approved in Apple's iTunes store. The company lists a set of guidelines and regulations app developers must achieve before they become available for purchase.

"They're not trying to be mean, they're trying to push you to do a better job--and they did," Feraco said of himself.

With persistence and practice comes success, however, and Feraco said it's much easier for him to get his apps approved now than when he started. He now has more than 15 apps in the iTunes marketplace. Many teachers have begun asking if he can create apps for them--and so far he's

been able to [take requests](#) for a small start-up fee.

Feraco also has a [Facebook page](#) filled with resources, like [this link to DailyTekk's article](#) on useful app development tools.

Despite the difficulty of learning how to create one's own apps, Feraco said that if teachers are willing to put forth the necessary effort the process is rewarding.

"If you're up for a challenging, exciting field, I highly recommend you learn about computer science and get started," he said.