

Syllabus - Teaching App Creation with MIT App Inventor

Course goal:

To enable adults (mostly teachers) taking the course to work with students to build apps using App Inventor (in a variety of potentially interdisciplinary settings, but definitely including App Challenge teams who want to complete their app).

Sub-goals:

To achieve the Course Goal, the course is designed with three strands, each of which has its own sub-goal:

- *App (Software) Design Process*
How to effectively design and develop a concept and from it, an app (or other software).
- *Programming with App Inventor*
How to use App Inventor as a tool within the design process; how to apply concepts & skills that will eventually allow build-out of an app idea.
Remember, for many of the topics we will cover in the programming strand of the course, the goal is not complete mastery of the topic, but knowing what App Inventor can do, and how to get more information about particular topics when/if it becomes necessary. We hope to prepare you to learn with students, not make you THE App Inventor expert.
- *Pedagogical Approaches*
How to teach with technology; how to teach design; how to teach programming
In this strand especially, we anticipate a variety of experiences, from experts in teaching with technology to relative novices. Because of this, this strand will be mostly reflection based.

How to get the most out of this course:

- *Don't be Afraid to Try (and sometimes try again)*
Some participants will be pushed to expand their comfort zone by this course. That's OK and expected - please try things, even if you're not quite sure what you're doing is right. You won't break anything, and chances are you'll figure out what to do and learn something in the process.
- *Have Fun!*
As teachers, you know that learning is inherently rewarding, often hard, but ultimately fun too. Remember that, and do your best to have fun as you travel through the course.

Time Commitment:

- *About 6 hours per unit (or week)*
This will vary by participant, but the course was designed to take about 6 hours per week for each of the six weeks or units (with one unit/week) that the course runs.

Course Structure:

Each week, we will cover one unit of material. Each unit includes a piece or pieces from each of the three strands described above (Design, Programming, Pedagogy). The design portion of each week will include a short reading explaining the topic covered, and an activity to help better understand that topic. The programming portion of each week will cover one or more concepts and skills, with short assignments to practice or apply those. The pedagogy portion of a week may include a brief reading, but will mostly focus on reflection questions to foster thoughtful

engagement. The topics covered in each unit will be:

- **Unit 1**
 - Programming - event handling, debugging
 - Design - overview of design process
 - Pedagogy - constructivism, constructionism, and inquiry-based learning;
- **Unit 2**
 - Programming - conditionals, lists, iteration
 - Design - need-finding
 - Pedagogy - progression of teaching with technology; interdisciplinary computing
- **Unit 3**
 - Programming - procedures, variables, algorithms, randomness
 - Design - prototyping
 - Pedagogy - teacher's role in learning process
- **Unit 4**
 - Programming - persistent data
 - Design - whittling ideas down to a chosen few
 - Pedagogy - standards (CSTA & College Board Computer Science Principles); Personal Learning Networks (PLNs)
- **Unit 5**
 - Programming - fusion tables, online APIs
 - Design - user testing
 - Pedagogy - best practices for teaching in a computing environment
- **Unit 6**
 - Programming - activity starter, layout & visual design, (other topic(s) as needed by participants)
 - Design - usability and UX
 - Pedagogy - lesson plan design; coding vs. computational thinking vs. computing vs. computer science - oh my!

Why Should You Participate?

- Learning is fun!
- This is something you could use in your school or classes - app programming is interdisciplinary - it can fit into almost any subject.