MYCS: COMPUTER SCIENCE FOR BEGINNERS

Course Information and Syllabus

WHAT IS MYCS?

What is computing? What do computer scientists do? How do computers "think" and deal with information? What is programming, and how do we use it?

MyCS, or "Middle-years Computer Science," is an early introduction to computer science (CS) concepts designed especially for middle school students, as well as those slightly older or younger. Because of the intended audience, the course tries to introduce the big ideas of CS without relying on a lot of math or computer skills to do so. It's a fun way to build a foundational understanding of what CS is all about, making it easier for you to take more advanced CS and programming courses when you've finished.

Computer science can be creative, challenging, and fun for anyone. We hope that every student finishes this course with the belief that "**CS is something that people like me can do,**" regardless of any previous background in computing.

COURSE STRUCTURE

MyCS: Computer Science for Beginners is composed of five units of curriculum, which alternate back and forth between the ideas that shape computer science and Scratch programming activities. The topics of these units are listed below.

Unit 1: What is Computer Science?
Answer broad questions about the role of computers and the goals of computer scientists.
Explore the definition of intelligence as it relates to computers.

Unit 2: A-maze-ing Scratch
Learn the basics of Scratch programming through a series of pre-made mazes of increasing difficulty.

Unit 3: Data and Codes
Practice encoding and decoding information using a variety of codes and methods.
Learn to represent numbers in binary.
Connect these concepts to computer science and working with data.

Unit 4: Projects in Scratch
Create your own stories, games, and interactions using Scratch.
Practice design skills for making unique programming projects.

Unit 5: Problem Solving and Algorithms
Build intuition for how people and computers solve problems differently
Learn basic algorithms for searching and sorting information, as well as how we can compare these algorithms.

After completing Unit 5, students wishing for additional practice in Scratch should consider signing up for CS002x: Programming in Scratch. While early parts of this Scratch programming course will be review, it will also explore in much more depth how problem solving and algorithms can be used within Scratch.

SCHEDULE

This course is self-paced. We recommend that a relaxed pace for the course is to take it over a five-week period, pursuing one unit per week, with an investment of only a few hours per week. However, the course may be started at any time after the start date of **May 12th, 2015, 0:00 UTC** and pursued at any pace.

If you wish to receive a certificate in this course, a minimum grade of **65%** must be achieved by the course's end date of **October 4th, 2015, 23:59 UTC**. Those looking to complete a *Verified Certificate* will need to upgrade by **September 25th, 2015, 23:59 UTC.**

The courseware will not be closed off after the course end date; any registered users of the course will be able to access the archived courseware indefinitely, including videos and assessments. Discussion boards will not be moderated after the course end date.

> *A note for teachers:* If you wish to retain access to the curriculum after October 4th, please remain enrolled in the course, as this version of the course will not be available for additional registrants if a new version of the course opens for registration.

NOTES ABOUT THIS COURSE

This isn't a normal MOOC.

If you're a veteran MOOC-taker, this course probably won't look like previous classes you've taken. It's not a college-level course, and it's not structured to look like one. So, before you get going, we want to make sure you have some idea of what to expect from this MOOC.

VIDEOS

The videos in this course were made by undergraduate students to help explain concepts and activities to young students. There aren't as many videos as you might expect in more advanced CS courses, nor are the videos as high in production quality as other series you might see.

The goal is to be silly and attention-grabbing, as well as to provide one way of explaining the main elements of the course. **If they're not your style, don't worry** - you can still learn everything in this course without using the videos.

That said, if you (or your students, if you're a teacher) want to make a more in-depth explanation of a concept, whether using pictures, text, or especially video, we encourage you to do so! Teaching can be a great way to learn, and we always welcome new explanations and opportunities to improve our curriculum. You can share your creations in our discussion boards.

GRADING

In the courseware, you'll have plenty of opportunities to check your understanding - mini-quizzes, assignments, questions, and opportunities for discussion. However, you won't ever be required you to do anything that looks like an exam. You'll never have a limited number of opportunities to complete an assignment, and won't be graded on your ability to recite all of the content for a given unit.

This can make the course a lot easier than others you've taken in some ways, but it also means that it's on you to decide what you want to take away from the course. If you want to review anything, we encourage you to go back and look through the content again. Plus, you can always review the archived content when the course finishes up.

TEACHER RESOURCES

This course is especially for teachers. We designed this course especially with an eye for teachers who are relatively new to computer science, but would like to teach CS concepts to younger students. Especially in the United States, it can often be challenging to find opportunities to learn about CS even in high school, let alone earlier. However, learning about it early on gives students more time to build their interests and skills, as well as to see the many reasons this field can be more fun and compelling than you might think.

So, with that in mind, the lessons taught within this MOOC will be supplemented by Teacher Resources, which will provide a curriculum resource to expand on this MOOC courseware for up to a full semester of lesson plans. The MOOC serves as a "highlight reel" for this full curriculum, fleshing out the lessons that are often the most interesting to students.