Project Goals

Computer Science Matters in Maryland (“CS Matters”) is an NSF-supported effort to increase the availability and quality of high school CS courses across the state of Maryland. The project is led by Dr. Marie desJardins (UMBC), Dr. Jan Plane (UMCP), Ms. Dianne O’Grady-Cunniff (Westlake High School, Charles County), Mr. Joe Greenawalt (La Plata High School, Charles County), and Ms. Christina Morris (Catonsville High School, Baltimore County).

In 2014, we assembled a cohort of master teachers (from nine public and private school systems in Maryland and the District of Columbia) to create a complete curriculum package for the new College Board CS Principles Advanced Placement (AP) course. Our master teachers will pilot the new curriculum in the 2014-15 school year. In 2015, we intend to offer two workshops (one at UMBC and one at UMCP) to train 30 additional pilot teachers on the new curriculum, and support them as they teach the course during the 2015-16 school year. In 2016, we aim to partner with four training sites across the state of Maryland and the District of Columbia, training 80 additional teachers to teach the course during the 2016-17 school year (the first year in which the AP CS Principles exam is scheduled to be offered).

More information about CS Matters in Maryland is available at the project website, http://csmatters.org. For questions or feedback, please contact the project team at csmattersinmaryland@gmail.com or PI Marie desJardins at mariedj@cs.umbc.edu.

Curriculum Overview

The CS Matters in Maryland CS Principles AP course incorporates a focus on active, inquiry-based learning. The structure of the course is designed to meet all of the CS Principles learning objectives, to prepare the students for the three CS Principles performance tasks, and to spread out the work on these tasks over the course of the year. The overarching theme of the course is data: the nature and variety of data on the Internet; algorithmic methods for processing and managing data; and ways in which data can be analyzed, visualized, and interpreted to increase human understanding and solve challenging real-world problems. Programming concepts are taught using Python. The six units, and the three performance tasks, are organized as follows:

Unit 1: Your Virtual World
Unit 2: Information and the Internet
Explore: Research an innovation that uses the Internet

Unit 3: Developing Programs
During the curriculum development process, master teachers worked in small “unit teams” to create the lessons within each unit. Each lesson was then reviewed by five independent review panels to ensure that the curriculum met our design criteria: (1) consistency with and complete coverage of the CS Principles learning objectives, (2) alignment with Common Core and NGSS standards, (3) a focus on active learning pedagogies, (4) differentiation for diverse student populations and different learning styles, and (5) college-level, rigorous content. Many lessons include extensions and variations to facilitate the course’s adoption in different contexts and for different student populations. The course is designed to fit within one hundred fifty 50-minute “sessions,” but we recognize that many schools have a block schedule or other scheduling method. Our explicit goal was to design the lessons with as much flexibility as possible.

We are grateful to Code.org for sharing their curriculum, which was in development as we began the creation of our own curriculum. We have borrowed and adapted many of the individual lessons from Code.org’s curriculum. Like their curriculum, the CS Matters in Maryland CS Principles AP course will be distributed under a Creative Commons ShareAlike license, and can be adapted or reused with attribution for non-commercial purposes. We anticipate releasing the first public version of the CS Matters CS Principles curriculum in January 2015. The summary below is tentative and subject to change.

1. Your Virtual World
The main idea of Unit 1 is to explore the effect of data on students’ lives. Students will discover what is known about them online, explore the issue of data privacy, and investigate their rights to access or change that data. After finding ways that the world of information is changing, students will learn how data is stored using binary codes, hardware, and files.

Unit 1 Lessons:

1. The Impact of Innovation on Your Life
2. Representing Bits
3. Number Conversions
4. The Impact on Society and Other Fields
5. Smart Searching
6. Privacy in the Age of Big Data
7. Unequal Access to Computing
8. A Problem Solving Process that Scales
2. Information and the Internet
The main idea of Unit 2 is to explore the Internet to prepare students to do research on a current Internet-related topic in detail on their own. A variety of algorithms from different areas of interest are investigated. Students will learn how information is transmitted online and how search engines find and organize the data using complex algorithms. Studying cybersecurity and cryptography will allow students to understand more about the issues concerning the privacy of data online, leading to a discussion about the ethical and social issues surrounding the increased use of online data.

Unit 2 Lessons:
1. Information Transmission and the Web
2. How the Internet Works
3. Algorithms
4. How Search Engines Work
5. Cybersecurity
6. Cryptography
7. Cloud Computing and Ethics
8. How Innovation Impacts Our Lives

Performance Task: Explore. Research an innovation that uses the Internet.

3. Developing Programs
The main idea of Unit 3 is to create solutions with code, debug these programs, and verify their results. Programming uses creative expression to solve problems correctly using abstraction, both individually and collaboratively. Students will implement algorithms using math and logic, and evaluate programs for correctness.

The programming unit takes the formality of algorithmic expression one step further by having students write programs in Python. In the programming unit, the levels of abstraction increase as students learn how a programming language can be used to control the machine on which the program is running. The topics of the programming unit are input/output, calculations with numbers, branching statements and Boolean logic, iteration, procedural abstraction, and processing data in a list.
We plan to use Python 3.4.1 with the PyCharm IDE and the textbook *How to Think Like a Computer Scientist*, hosted by Runestone Interactive: 
http://interactivepython.org/runestone/static/CCPS_Python/index.html

**Unit 3 Lessons:**

1. Motivation for Creating Programs
2. Input, Output, and Debugging
3. Expressions, Numbers, and Strings
4. Conditional Statements and Logic
5. Iteration
6. Functions, Parameters, and Abstraction
7. Manipulating Strings
8. Lists

**4. Data Acquisition**
The main idea of Unit 4 is to enable students to find data online and analyze it using a spreadsheet and Python code. To prepare for the performance task, students will follow the pattern of developing a hypothesis and then exploring and analyzing data to find the answer. File input and output will allow students to use large sets of data, find patterns, look for metadata (data about data), and create visualizations of the data. Students will also investigate what is a reasonable solution and how the new data revolution can drive discovery and decision making.

**Unit 4 Lessons:**

1. File Input and Output
2. Properties of Number Collections
3. Basic Statistics
4. Testing a Hypothesis with Real Data
5. Finding Patterns and Looking for Metadata
6. Data Visualizations
7. Comparing Solutions
8. Discovery and Decision Making Through the Data Revolution

---

*Performance Task: Investigate.* Find and analyze your own data.
5. Data Manipulation
The main idea of Unit 5 is to create a synthesis of algorithms, programming, and data. Python lists will be used to work with a large set of data using searching, sorting, and other ways to manipulate data. Models and simulations will be used to represent real-life situations. Algorithms will be compared for their effectiveness as well as their readability.

Unit 5 Lessons:

1. Working with Big Data
2. Searching
3. Sorting
4. Models and Simulations
5. Comparing Algorithms
6. Advanced Algorithms
7. Group Project

6. Data Visualization
The main idea of Unit 6 is to explore ways to create and understand data through data visualization. Students will also prepare for the final performance task by doing a group project.

Unit 6 Lessons:

[Under development.]

Performance Task: Create (Write a program to analyze data graphically.)