

Introduction to Computer Science Course Proposal

Grade Level, duration, number of students:

- Mathematics elective with open enrollment for grades 9 - 12
- Semester-long course to be run during each semester
- Optimal enrollment would be 18 students with a maximum of 24 students

Statement of Need:

At the Junior/Senior High School we currently have two vastly different course offerings under the umbrella of computer science: *Technology 8* and *AP Computer Science*. The former briefly introduces elementary forms of computer programming and coding. The latter requires a deep investigation of the subject. The inherent gap between these courses necessitates a course to act as a bridge for those who choose to take AP Computer Science, while providing a developmentally appropriate learning environment for those who have a keen interest in the field but are not ready for the complexity of a college-level computer science course.

Statement of Goals:

The creation of this course will provide students with a strong understanding of what computer science is. Students will explore a variety of programming languages, broaden their understanding of logic and algorithms, and increase their awareness of the applicability of the subject field in the modern world.

Objectives:

- Meet the CSTA (Computer Science Teachers Association) standards as outlined at http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_Standards_Mapped_to_CommonCoreStandardsNew.pdf
- Provide students with alternative options to explore in relation to computer science
- Engage students in real-world, problem-solving, computer challenges
- Empower students with a much needed command of functions and data manipulation contained within spreadsheets
- Develop advanced concepts such as loops and data structures
- Facilitate an understanding of the differences between computer skills and programming
- Serve as a possible pathway to AP Computer Science

The Course (a brief course outline):

The Introduction to Computer Science course is a hands-on, thought provoking course that will teach students how to read, speak, and think like a computer scientist. The course will cover the fundamentals of programming through different media, and allow for the creation of functional programs. The course will also introduce much needed skills involving cell functions, error-checking, data validation and manipulation, and analysis. Lastly, the course will give a basic understanding in how to write beginner code in HTML, CSS, and Java. Because the world of programming has changed dramatically over the past twenty years, this course will constantly be changing to meet the demands of new and superior technology.

Course of Sequence Outline:

Introduction to Computer Science Course Outline	
Unit 1	Scratch <ul style="list-style-type: none"> ❖ Exploring ❖ Animations ❖ Stories ❖ Diving Deeper
Unit 2	Using Logic <ul style="list-style-type: none"> ❖ Problem solving ❖ Conditionals ❖ Truth tables
Unit 3	Maximizing spreadsheets with programming <ul style="list-style-type: none"> ❖ Basic cell functions ❖ Looping ❖ Error Checking ❖ Data manipulation ❖ Advanced cell functions
Unit 4	Searching and Sorting <ul style="list-style-type: none"> ❖ Sequential search ❖ Binary search ❖ Hash tables ❖ Linear sort
Unit 5	Introduction to Algorithms <ul style="list-style-type: none"> ❖ Measures of efficiency ❖ Shortest path problems ❖ Traveling salesman problems
Unit 6	Exploring Number Systems <ul style="list-style-type: none"> ❖ Binary ❖ Hexadecimal ❖ Octal

Unit 7	Webpage development using HTML and CSS <ul style="list-style-type: none"> ❖ Fundamentals ❖ Frames, Headers, Links ❖ Lists ❖ Tables ❖ Color ❖ Buttons
Unit 8	Programming in Java <ul style="list-style-type: none"> ❖ Hello World ❖ Interactivity ❖ If statements

Instructional Techniques:

Computer science is best learned with hands-on, activity-based learning opportunities. Students will work individually to learn the aspects of computer science, and in groups to analyze, synthesize, brainstorm and form solutions to given problems. The course will also include project-based learning opportunities, where students will routinely work together to learn and complete projects.

Using technology will become a daily practice in the class. The scratch learning community will be utilized to allow students to create interactive stories and games while learning the basic fundamentals of computer science. HTML and CSS will be incorporated on the computer so students can see web pages they develop. The beginner Integrated Development Environment (IDE) BlueJ will be utilized to explore the Java programming language.

Material and Resources:

Brennan, K., Balch, C., Chung, M., (2014). Creative Computing. Cambridge, MA: Harvard Graduate School of Education. Available at <http://scratched.gse.harvard.edu/guide/>

Evaluation Procedures:

Students will be evaluated in multiple ways: through test/quiz completion, small & large group activities, projects & presentations, and through homework completion. Students will be encouraged to discuss possible solutions to a given problem in a group setting.

The course will culminate with a final project that will be based on the concepts covered throughout the course. Students will be given the option to complete a project based on one of the many units of study, while paralleling their own personal interests.

Estimated Space and Personnel:

This course would require a computer lab or a designated space with a dedicated laptop cart. Each computer/laptop would need an IDE, preferably BlueJ to be able to run Java programs. The computers would also need to readily access the internet.

Estimated Direct Costs:

- TBD (Dependent on computer access in relation to room location/period course is scheduled for)

Curriculum Writing Time:

20 hours