El Monte Union High School District

Revised 01-08-2018

Course Outline

District: EMUHSD
High School: DISTRICT

Course Title: Introduction to Computer Science with JavaScript

Textbook(s): (web-based)

Copyright date/Edition:

Transitional*_____(Eng. Dept. Only)

Sheltered (SDAIE)*___Bilingual*___

AP**_____Honors**_____

Department: Business

CTE*** :
Industry Sector:_____
Pathway:_____

Check One Introductory:
_____X_____
Concentrator:_____
Capstone:_____

Grade Level (s): 9-10

Semester X_____Year_____

Year of State Framework Adoption_____

This course meets graduation requirements:

( ) English
( ) Fine Arts
( ) Foreign Language
( ) Health & Safety
( ) Math
( ) Physical Education
( ) Science
( ) Social Science
(X) Elective

This course meets a-g requirements:

( ) "a" – Social Studies
( ) "b" – ELA
( ) "c" – Math
( ) "d" – Lab Science
( ) "e" – Language (not English)
( ) "f" – Vis/Perf Arts
( ) "g" – College prep elective

Department/Cluster Approval Date

_____ _____

_____ _____

_____ _____

_____

_____ _____

_____ _____

_____

Is this course an adaptation from another source?

- ☐ No
- ☐ Yes

If yes, please indicate the source of the original course:

*Instructional materials appropriate for English Language Learners are required.

For AP/Honors course **attach a page describing how this course is above and beyond a regular course. Also, explain why this course is the equivalent of a college level class.

***For CTE, **attach the CTE course outline** created in the online template (http://ctecourse.scoe.net/).

1. **Prerequisite(s):**

The Intro to Computer Science in JavaScript course is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for new coders.

2. Short description of course which may also be used in the registration manual:

- **Objectives of course**
  The introduction to computer science curriculum teaches the foundations of computer science and basic programming, with an emphasis on helping students develop logical thinking and problem solving skills. Once students complete the Introduction to Computer Science course, they will have learned material equivalent to a semester college introductory course in Computer Science and be able to program in JavaScript.

- **3-5 sentences explaining overall course content**
  The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total.

- **Indicate references to state framework(s)/standards (If state standard is not applicable then national standards should be used)**

  California English Common Core Standards
  RST.9-10.3 Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
  RST.11-12.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11—12 texts and topics.
  WHST.11-12.2 Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.

  California Math Common Core Standards
  5.G.1 Use a pair of perpendicular number lines, called axes, to define a coordinate system, with the intersection of the lines (the origin) arranged to coincide with the 0 on each line and a given point in the plane located by using an ordered pair of numbers, called its coordinates. Understand that the first number indicates how far to travel from the origin in the direction of one axis, and the second number indicates how far to travel in the direction of the second axis, with the convention that the names of the two axes and the coordinates correspond (e.g., x-axis and x-coordinate, y-axis and y-coordinate).
  6.NS.8 Solve real-world and mathematical problems by graphing points in all four quadrants of the coordinate plane. Include use of coordinates and absolute value to find distances between points with the same first coordinate or the same second coordinate.
  A.CED.2 Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales.

  California Academic Content Standards
  ELA.9-10.W.2.1b Write biographical or autobiographical narratives or short stories that locate scenes and incidents in specific places.
  ELA.9-10.W.2.4c Write persuasive compositions that clarify and defend positions with precise and relevant evidence, including facts, expert opinions, quotations, and expressions of commonly accepted beliefs and logical reasoning.
  ELA.9-10.W.RT.1.8 Design and publish documents by using advanced publishing software and graphic

programs.
ELA.9-10.W.RT.1.3 Use clear research questions and suitable research methods (e.g., library, electronic media, personal interview) to elicit and present evidence from primary and secondary sources.
ELA.9-10.R.CAGT.2.6 Demonstrate use of sophisticated learning tools by following technical directions (e.g., those found with graphic calculators and specialized software programs and in access guides to World Wide Web sites on the Internet).
ELA.9-10.R.CAGT.2.3 Generate relevant questions about readings on issues that can be researched.
ELA.9-10.R.CAGT.2.5 Extend ideas presented in primary or secondary sources through original analysis, evaluation, and elaboration.
ELA.11-12.W.OF.1.1 Demonstrate an understanding of the elements of discourse (e.g., purpose, speaker, audience, form) when completing narrative, expository, persuasive, or descriptive writing assignments.
M.8-12.A.2.0 Students understand and use such operations as taking the opposite, finding the reciprocal, taking a root, and raising to a fractional power. They understand and use the rules of exponents.
M.8-12.A.5.0 Students solve multistep problems, including word problems, involving linear equations and linear inequalities in one variable and provide justification for each step.
VA.3.HCC.RDVA.3.3 Distinguish between and describe representational, abstract, and nonrepresentational works of art.

California's 2013 CTE Standards
CTE.ICT.A.6.2 Use a logical and structured approach to isolate and identify the source of problems and to resolve problems.
CTE.ICT.D.3.2 Create a design specification document to include interface and delivery choices, rules of play, navigation functionality, scoring, media choices, start and end of play, special features, and development team credits.
CTE.ICT.D.3.3 Using simple game development tools, create a game or simulation.
CTE.ICT.D.1.7 Identify the core tasks and challenges that face a game or simulation design team.
CTE.ICT.D.1.4 Describe the psychological impact of games on individuals and groups.
CTE.ICT.D.2.6 Demonstrate an understanding of the techniques used to evaluate game mechanics, game play, flow, and game design.
CTE.ICT.KPAS.11.1 Utilize work-based/workplace learning experiences to demonstrate and expand upon knowledge and skills gained during classroom instruction and laboratory practices specific to the Information and Communication Technologies sector program of study.
CTE.ICT.KPAS.11.2 Demonstrate proficiency in a career technical pathway that leads to certification, licensure, and/or continued learning at the postsecondary level.
CTE.ICT.KPAS.5.5 Use a logical and structured approach to isolate and identify the source of problems and to resolve problems.
CTE.ICT.KPAS.5.8 Create and use algorithms and solve problems.
CTE.ICT.KPAS.10.14 Analyze the effectiveness of online information resources to support collaborative tasks, research, publications, communications, and increased productivity.
CTE.ICT.KPAS.10.1 Interpret and explain terminology and practices specific to the Information and Communication Technologies sector.
CTE.ICT.KPAS.10.11 Know multiple ways in which to transfer information and resources (e.g., text, data, sound, video, still images) between software programs and systems.
CTE.ICT.KPAS.4.4 Discern the quality and value of information collected using digital technologies, and recognize bias and intent of the associated sources.
CTE.ICT.KPAS.4.1 Use electronic reference materials to gather information and produce products and services.
CTE.ICT.KPAS.4.5 Research past, present, and projected technological advances as they impact a particular pathway.
CTE.ICT.KPAS.8.7 Conform to rules and regulations regarding sharing of confidential information, as determined by Information and Communication Technologies sector laws and practices.

CTE.ICT.KPAS.8.8 Identify legal and ethical issues that have proliferated with increased technology adoption, including hacking, scamming, and breach of privacy.
CTE.ICT.KPAS.8.4 Explain the importance of personal integrity, confidentiality, and ethical behavior in the workplace.
CTE.ICT.KPAS.8.3 Demonstrate ethical and legal practices consistent with Information and Communication Technologies sector workplace standards.
CTE.ICT.KPAS.8.1 Access, analyze, and implement quality assurance standards of practice.
CTE.ICT.KPAS.2.2 Identify barriers to accurate and appropriate communication.
CTE.ICT.KPAS.3.4 Research the scope of career opportunities available and the requirements for education, training, certification, and licensure.
CTE.ICT.KPAS.3.9 Develop a career plan that reflects career interests, pathways, and postsecondary options.
CTE.ICT.KPAS.3.1 Identify personal interests, aptitudes, information, and skills necessary for informed career decision making.
CTE.ICT.C.5.4 Test software and projects.
CTE.ICT.C.3.2 Design effective and intuitive interfaces using knowledge of cognitive, physical, and social interactions.
CTE.ICT.C.6.1 Identify the basic design elements necessary to produce effective print, video, audio, and interactive media.
CTE.ICT.C.4.6 Use proper programming language syntax.
CTE.ICT.C.4.1 Identify and describe the abstraction level of programming languages from low-level, hardware-based languages to high-level, interpreted, Web-based languages.
CTE.ICT.C.4.9 Create programs using control structures, procedures, functions, parameters, variables, error recovery, and recursion.
CTE.ICT.C.4.5 Demonstrate awareness of various programming paradigms, including procedural, object oriented, event-driven, and multithreaded programing.
CTE.ICT.C.4.3 Identify and use different authoring tools and integrated development environments (IDEs).
CTE.ICT.C.4.4 Identify and apply data types and encoding.
CTE.ICT.C.4.11 Document development work for various audiences, such as comments for other programmers, and manuals for users.
CTE.ICT.C.4.8 Use object oriented programming concepts, properties, methods, and inheritance.
CTE.ICT.C.4.7 Use various data structures, arrays, objects, files, and databases.
CTE.ICT.C.1.1 Identify the phases of the systems development life cycle, including analysis, design, programming, testing, implementation, maintenance, and improvement.
CTE.ICT.B.5.1 Classify and use various electronic components, symbols, abbreviations, and media common to network topology diagrams.
CTE.AME.D.5.6 Test a classmate's game project and create a bug report for the game. For each error submitted, write steps in sufficient detail so it is identifiable and reproducible to the developer. Use a metric to identify how critical the error is based on its negative impact on game play.
CTE.AME.D.4.4 Create 2-D art and 3-D models.
CTE.AME.D.4.3 Explain how to create the illusion of 3-D in a 2-D environment.
CTE.AME.D.2.0 Analyze the core tasks and challenges of video game design and explore the methods used to create and sustain player immersion.
CTE.AME.D.6.1 Identify processes of design and development from concept to production, including content creation, filling team roles, design documentation, communication, and scheduling for video game design teams.

- **Student performance standards**

Demonstrate a willingness to learn.
Participate actively as a member of a team.
Communicate professionally with others through verbal, non-verbal and/or written communication.

Learn how the career path in Computer Programming and Computer Science can lead to rewarding and satisfying jobs in the future.
Study jobs in the future.
Describe the concept of object-oriented programming (OOP)
Locate and describe the components of the CodeHS interface
Complete the necessary online lessons
Print the code for methods and events
Understand the software development cycle and its four phases
Describe the different types of software documentation and their appropriate uses
Understand modular development and its benefits
Understand top-down software design and bottom-up software development
Perform unit testing and integration testing
Create a compound Boolean expression in a looping or branching instruction
Describe the concept of an event in computer programming
List and briefly describe the nine event types used
Add an invisible object to mark a spot
Create a new Boolean and numeric properties for an object
Create an event to move an object
Design, create, troubleshoot and play simple video games.
Assess, and offer constructive suggestions for others' projects.
Work in collaborative settings to design and create video games.

- **Evaluation/assessment/rubrics**
  Each unit ends with a comprehensive unit test that assesses student's mastery of the material from that unit. The Final Exam will consist of students creating their own interactive games using a combination of timers, randomization, mouse events, keyboard events, functions, graphics, and animation.

- **Include minimal attainment for student to pass course**
  Students must complete units 1, 3, 4, 5, and 6 to receive a passing grade for the course.

3. Course content:

Number of units (minimum of 6): 8

Unit 1: Introduction to Programming in JavaScript with Karel the Dog (6 weeks/30 hours)

Browse the full content of this unit at https://codehs.com/library/course/1/module/1

Objectives / Topics Covered	● Commands ● Defining vs. Calling Methods ● Designing methods ● Program entry points ● Control flow ● Looping ● Conditionals ● Classes ● Commenting code ● Preconditions and Postconditions ● Top Down Design
Assignments / Labs	● 30 Karel Programming Exercises and Challenges in total ● Program-specific tasks for Karel the Dog ○ Example Exercise: Pyramid of Karel Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row. ● Teach Karel new commands like `turnRight()` or `makePancakes()` ○ Example Exercise: Pancakes Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th avenue. Each stack of pancakes should have three pancakes. Create a method called `makePancakes()` to help Karel solve this problem. ● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design ○ Example Exercise: The Two Towers In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. At the end, Karel should end up on top of the second tower, facing East. ● Using control structures and conditionals to solve general problems ○ Example Exercise: Random Hurdles Write a program that has Karel run to the other side of first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long. ○ Example Exercise: Super Cleanup Karel Karel's world is a complete mess. There are tennis balls all over the place, and you need to clean them up. Karel will start in the bottom left corner of the world facing east, and should clean up all of the tennis balls in the world. This program should be general enough to work on any size world with tennis balls in any locations.

Unit 2: Karel Challenges (optional) Should be completed by Advanced students within the 1st 6 weeks
Browse the full content of this unit at https://codehs.com/library/course/1/module/1407

Objectives / Topics Covered	● Solving large and more complex problems using Karel
Assignments / Labs	● Several Karel challenges to tie everything learned in the Karel module together

Unit 3: Javascript & Graphics (2 weeks/10 hours)
Browse the full content of this unit at https://codehs.com/library/course/1/module/2

Objectives / Topics Covered	● Variables ● User Input ● Arithmetic Expressions ● Booleans
Assignments / Labs	● Using variables and getting user input using JavaScript ○ Example Exercise: Grocery Store Prompt the user for their name, and then how many apples, and then how many oranges they would like to buy. Then print out the name that was given, as well as how many apples and oranges they wanted.

Unit 4: JavaScript Control Structures (4 weeks/20 hours)
Browse the full content of this unit at https://codehs.com/library/course/1/module/1410

Objectives / Topics Covered	● Booleans ● For Loops ● Conditionals ● Nested Control Structures ● While Loops
Assignments / Labs	● Using comparison and logical operators to control the flow of the program ○ Example Exercise: Inventory Write a program that keeps track of a simple inventory for a store. While there are still items left in the inventory, ask the user how many items they would like to buy. Then print out how many are left in inventory after the purchase. You should use a while loop for this problem. Make sure you catch the case where the user tries to buy more items than there are in the inventory. In that case, you should print a message to the user saying that their request isn't possible.

	● Using for loops ○ Example Exercise: All Dice Values ■ Write a program that prints all possible dice rolls with 2 dice. To do so, you should use a double for loop. Hint: You can't use i for both for loops. ● Drawing basic graphics using JavaScript ○ Example Exercise: French Flag This program should draw the French flag. The left third of the canvas is blue, the middle third is white, and the right third is red. You will need to use Rectangle objects in this program. ○ Example Exercise: Caterpillar This graphics program should draw a caterpillar. A caterpillar has NUM_CIRCLES circles. Every other circle is a different color, the even circles are red, and the odd circles are green (by even we mean when i is an even number). Use a for loop to draw the caterpillar, centered vertically in the screen. Also, be sure that the caterpillar is still drawn across the whole canvas even if the value of NUM_CIRCLES is changed.

Unit 5: Functions and Parameters (4 weeks/20 hours)

Browse the full content of this unit at https://codehs.com/library/course/1/module/1410

Objectives / Topics Covered	● Functions with and without parameters ● Functions with and without return values ● Nested Control Structures ● Local variables and scope
Assignments / Labs	● Using various kinds of functions such as functions with and without parameters, and functions with and without return values ○ Example Exercise: Horizontal Lines Write a function that draws horizontal lines on the graphics canvas. If a line is horizontal, then the y-values for the endpoints are the same. The parameters to your function should be the y location, and the length, and all of your lines should start at x position 0. ○ Example Exercise: Is it even? Write a function called isEven that returns a boolean of whether or not a value is even or odd. The isEven function should not print anything out or return a number. It should only take in a number and return a boolean. Once you've written this function, write a program that asks the user for integers and prints whether the number they entered is even or odd using your isEven function. You should let the user keep entering numbers until they enter the SENTINEL given.

Unit 6: JavaScript and Graphics Challenges (1 week/5 hours) Revised 01-08-2018

Browse the full content of this unit at https://codehs.com/library/course/1/module/3

Objectives / Topics Covered	● Solving large and more complex problems using in JavaScript and graphics
Assignments / Labs	● Graphics Challenges to tie everything in the module together ○ Example Exercise: Ghosts Write a program to draw ghosts on the screen. You must do this by writing a function called drawGhost, which takes three parameters, the center x location of the ghost, the center y location of the ghost and the color of the ghost.

Unit 7: Animation and Games (Optional)

Browse the full content of this unit at https://codehs.com/library/course/1/module/3

Objectives / Topics Covered	● Timers ● Randomizing Games ● Mouse Events ● Keyboard Events
Assignments / Labs	● 13 exercises in total ● Using timers to add randomizations to graphical programs ○ Example Exercise: Paint Splatter Write a program that splatters paint on the screen every DELAY milliseconds. To splatter paint, pick a random color and draw CIRCLES_PER_SPLATTER circles of that color at random places on the screen. The radius of each circle should be a random value between MIN_RADIUS and MAX_RADIUS. Remember to use helper functions. ● Using mouse events for interactive programs ○ Example Exercise: Teleporting Ball Extend our bouncing ball program. Whenever you click, the ball should teleport to that spot and change to a random color. ○ Example Exercise: Target Draw a target on the screen that moves to aim at where your mouse is located. A target consists of a horizontal line that goes from 0 to the window width and a vertical line that goes from 0 to the window height. The lines should cross paths where the mouse is. If you're feeling adventurous, you can extend this to draw a small red circle whenever you click. If you're feeling really adventurous, you can have a bouncing ball on the screen and see if you can remove it when it gets clicked. You can use remove(obj) to remove something from the screen and getElementAt(x, y) to get an object at the given position. It will return the object or will return null if there is no object there. ● Using keyboard events for interactive programs

	○ Example Exercise: Basic Snake Write a basic version of the snake game. The way our game works is by first creating a green square at the center of the screen. The snake should be moving to the right. If you hit an arrow key, you should change the snake's direction.

Unit 8: Project: Breakout (Optional)

Browse the full content of this unit at https://codehs.com/library/course/1/module/469

Objectives / Topics Covered	● Basic graphics ● Mouse events ● Collision detection
Assignments / Labs	● Guided exercises to build a Breakout Game ● The Breakout Game is made up of bricks at the top of the screen, a paddle that you control at the bottom of the screen, and a ball that bounces around. Your goal is to direct the paddle with your mouse to bounce the ball until all of the bricks have been hit and disappear.

Unit 9: Optional Supplemental Materials (Remainder of school year)

Objectives / Topics Covered	● Extra practice with: ○ Karel ○ Basic JavaScript ○ JavaScript functions ○ Graphics ○ Animation ● Basic Data Structures in JavaScript ● Game design ● Music Visualization
Assignments / Labs	● Several additional exercises and large projects covering the topics listed above

4. **Describe how this course integrates the schools SLO (former ESLRs- Expected School-wide Learning Results):**

The following SLO's will be integrated into class work, assignments, labs, and projects.

Academic Schoolwide Learner Outcomes

• Seek, access, analyze, and creatively use information to demonstrate effective communication, computation,
critical thinking, and technological skills.

• Demonstrate proficiency in curricular programs aligned to the Common Core State Standards.
Interpersonal Academic Schoolwide Learner Outcomes

• Be productive community members by learning to respect diversity, exercise rights, accept responsibility, and
work cooperatively with others.

Personal Skills Schoolwide Learner Outcomes

• Make informed decisions, set goals, take actions, and evaluate results while exhibiting resiliency, honesty,
integrity, and personal accountability.

Career Student Learner Outcomes

• Explore a variety of career options and develop personal attributes and skills that lead to the pursuit of a post-
secondary education and productive work life.

5. **Describe the Integrated ELD teaching techniques to be used to meet the needs of English Language Learners:**

Students are instructed using various techniques such as small group discussions, mini lectures, computer presentations, demonstrations, YouTube videos, and one-on-one tutoring.

6. **Describe the interdepartmental articulation process for this course:**

Since the objectives of the Intro to Computer Science course expect students to read and write code, communicate, calculate, and document their work, students must apply their knowledge in language arts and math to complete assignments and projects. In addition, students are exposed to math concepts as they learn about computer programming.

7. **Describe how this course will integrate academic and vocational concepts, possibly through connecting activities. Describe how this course will address work-based learning/school to**

career concepts:
This class combines computer concepts with hands-on activities and simulation. Therefore, students are expected to integrate their academic knowledge with vocational principles to complete assignments and projects. Through the course of the year, students are able to make connection between what they learn in core areas and translate these knowledge to real life scenarios in career technical education classes, which would provide the relevancy for mastering core subjects.

8. Supplemental Materials of Instruction (Note: Materials of instruction for English Language Learners are required and should be listed below.)

Type of material (book, manual, periodical, article, website, primary source document, etc.)	**Author**	**Publisher**	**Edition/ Year**	**URL**	**Primary book, read in its entirety? (Y/N)**
CodeHS				https://codehs.com/	
Youtube				https://www.youtube.com/	
Google				https://www.google.com/	
Codecademy				www.codecademy.com	
Code.org				https://code.org/	
Alice				http://www.aliceprogramming.net/	