SAN LEANDRO UNIFIED SCHOOL DISTRICT
COURSE PROPOSAL: Course of Study

SECTION A. COVER PAGE

1. Course Title: **Introduction to Computer Science**	7. Action □New Course ☒ Course Revision □ Title Change Only
2. **Date Submitted:** 11/16/16	8. Grade Level(s) □ 6 ☒9 ☒11 □ 7 ☒10 ☒12 □ 8
3. **Transcript Title/Abbreviation(s)**	9. Prerequisites (Please list.) □ **Required** Algebra 1 □ **Recommended** Algebra 2
4. **Transcript Course Code/Course Number :**	10. Seeking Program Distinction? □ YES (Check one below.) ☒NO □Honors □ AP □Other
5. Subject Area Computer Science	11. Is this a CTE course? □ YES □ NO If YES, complete Section C
6. Department Career Technical Education	12. Previously approved by UC? □ YES ☒NO □ NA for 6th-8th Year approved Year removed
13. UC ELIGIBILITY Already eligible? □ NA for 6th-8th ☒YES □NO a □ b□ c□ d□ e□ f□ g☒ Proposing eligibility?□ NA for 6th-8th ☒YES □NO a □ b□ c□ d□ e□ f□ g☒ Please mark one for approved eligibility or proposed eligibility.	14. Unit Value/Course Credit/Length of Course □ 0.5 (semester or half-year ☒ 1.0 (one year equivalent) □ 2.0 (two-year equivalent) □ Other
15. School Contact and Course Developer Name: Anthony Keithley Phone: 510-618-4600 x 2236 Email: akeithley@slusd.us Position: Teacher	
16. Approval Signatures: Approval Date: Department Chair 11/16/16 Principal: 11/16/16	DISTRICT OFFICE USE ONLY Date approved by Board Curriculum Committee 11/29/16 Date approved by Board

SECTION B. COURSE CONTENT

17. Course Description

This course is designed to introduce students with little or no prior knowledge of website development using html and programming using JavaScript. Throughout the course of the year, students will learn about the many of the html object types that can be used across different browsers as well as devices. Students will learn to construct a website and implement functionality through the use of JavaScript as well as many of the programming statements that are used in JavaScript and other programming languages such as C++, Visual Basic and Python. A strong mathematical background is a must as concepts such as logarithmic functions, trigonometric functions, sequences and series, modular arithmetic, as well as other Algebra 2 and Precalculus topics are frequently referenced.

The purpose of this course is to introduce students to computer programming along with many of the commands that are used in other computer languages such as C++, Visual Basic and Python. The goal of the course is for students to have an easier transition to other languages if the need arises that they need to learn another language for career needs.

The ultimate outcome will be for a student to be able to learn other programming languages by knowing what commands, variable types, events etc... are available for JavaScript and other computing languages. They will know how to research, collaborate and edit not only their own code, but the code of others as well.

18. Course Goals and/or Major Student Outcomes

The purpose of this course is to introduce students to computer programming along with many of the commands that are used in other computer languages such as C++, Visual Basic and Python. The goal of the course is for students to have an easier transition to other languages if the need arises that they need to learn another language for career needs.

The ultimate outcome will be for a student to be able to learn other programming languages by knowing what commands, variable types, events etc... are available for JavaScript and other computing languages. They will know how to research, collaborate and edit not only their own code, but the code of others as well.

19. Course Objectives (standards)

CTE ICT Standards Sections C
https://drive.google.com/a/slusd.us/file/d/0ByV56Bd34MGPTmlHbDVBcHpjNXc/view?usp=sharing

C1.0 Identify and apply the systems development process.
C1.1 Identify the phases of the systems development life cycle, including analysis, design, programming, testing, implementation, maintenance, and improvement.
C1.2 Identify and describe models of systems development, systems development life cycle (SDLC), and agile computing.

C1.3 Identify and describe how specifications and requirements are developed for new andexisting software applications.
C1.4 Work as a member of, and within the scope and boundaries of, a development project team.
C1.5 Track development project milestones using the concept of versions.
C1.6 Diagram processes using flowcharts and the Unified Modeling Language.

C2.0 Define and analyze systems and software requirements.
C2.1 Describe the major purposes and benefits of development, including automation, improving productivity, modeling and analysis, and entertainment.
C2.2 Recognize and prevent unintended consequences of development work: programming errors, security issues, health and environmental risks, and privacy concerns.
C2.3 Develop strategies that target the specific needs and desires of the customer.
C2.4 Analyze customers' needs for development.
C2.5 Determine and document the requirements and alternative solutions to fulfill the customers' needs.

C3.0 Create effective interfaces between humans and technology.
C3.1 Describe and apply the basic process of input, processing, and output.
C3.2 Design effective and intuitive interfaces using knowledge of cognitive, physical, and social interactions.
C3.3 Support methods of accessibility for all potential users, including users with disabilities and non-English-speaking users.

C4.0 Develop software using programming languages.
C4.1 Identify and describe the abstraction level of programming languages from low-level, hardware-based languages to high-level, interpreted, Web-based languages.
C4.2 Describe the interaction and integration of programming languages and protocols such as how client-side programming can work with server-side programming to use a query language to access a database.
C4.3 Identify and use different authoring tools and integrated development environments (IDEs).
C4.4 Identify and apply data types and encoding.
C4.5 Demonstrate awareness of various programming paradigms, including procedural, object oriented, event-driven, and multithreaded programing.
C4.6 Use proper programming language syntax.
C4.7 Use various data structures, arrays, objects, files, and databases.
C4.8 Use object oriented programming concepts, properties, methods, and inheritance.
C4.9 Create programs using control structures, procedures, functions, parameters, variables, error recovery, and recursion.
C4.10 Create and know the comparative advantages of various queue, sorting, and searching algorithms.
C4.11 Document development work for various audiences, such as comments for other programmers, and manuals for users.

C5.0 Test, debug, and improve software development work.
C5.1 Identify the characteristics of reliable, effective, and efficient products.
C5.2 Describe the ways in which specification changes and technological advances can require the modification of programs.
C5.3 Use strategies to optimize code for improved performance.
C5.4 Test software and projects.
C5.5 Evaluate results against initial requirements.

C5.6 Debug software as part of the quality assurance process.

C6.0 Integrate a variety of media into development projects.
C6.1 Identify the basic design elements necessary to produce effective print, video, audio, and interactive media.
C6.2 Describe the various encoding methods of media and trade-offs: vector graphics vs. bitmaps, and bit depth.
C6.3 Use media design and editing software: keyframe animation, drawing software, image editors, and three-dimensional design.
C6.4 Develop a presentation or other multimedia project: video, game, or interactive Websites, from storyboard to production.
C6.5 Analyze the use of media to determine the appropriate file format and level of compression.
C6.6 Integrate media into a full project using appropriate tools.
C6.7 Create and/or capture professional-quality media, images, documents, audio, and video clips.

C7.0 Develop Web and online projects.
C7.1 Identify the hardware (server) and software required for Web hosting and other services.
C7.2 Describe the full process of online content delivery, registering domain names, setting up hosting, and setting up e-mail addresses.
C7.3 Attract Web-site visitors through search engine optimization using various strategies like keywords and meta-tags.
C7.4 Enable e-commerce capabilities to sell products, create a shopping cart, and handle credit card transactions.
C7.5 Create an online project, Web-based business, and e-portfolio.
C7.6 Optimize fast delivery and retrieval of online content such as Web pages.

C8.0 Develop databases.
C8.1 Describe the critical function of databases in modern organizations.
C8.2 Identify and use the basic structures of databases, fields, records, tables, and views.
C8.3 Identify and explain the types of relationships between tables (one-to-one, one-to-many, many-to-many) and use methods to establish these relationships, including primary keys, foreign keys, and indexes.
C8.4 Use data modeling techniques to create databases based upon business needs.
C8.5 Use queries to extract and manipulate data (select queries, action queries).
C8.6 Develop databases that are properly normalized using appropriate schemas.
C8.7 Export and import data to and from other applications and a database recognizing the limitations and challenges inherent in the process.
C8.8 Analyze and display data to assist with decision making using methods like cross tabulations, graphs, and charts.

C9.0 Develop software for a variety of devices, including robotics.
C9.1 Demonstrate awareness of the applications of device development work, including personalized computing, robotics, and smart appliances.
C9.2 Install equipment, assemble hardware, and perform tests using appropriate tools and technology.
C9.3 Use hardware to gain input, process information, and take action.

C9.4 Apply the concepts of embedded programming, including digital logic, machine-level representation of data, and memory-system organization.
C9.5 Program a micro-controller for a device or robot.

C10.0 Develop intelligent computing.
C10.1 Describe models of intelligent behavior and what distinguishes humans from machines.
C10.2 Describe the major areas of intelligent computing, including perception, proximity, processing, and control.
C10.3 Know artificial intelligence methods such as neural networks, Bayesian inferences, fuzzy logic, and finite state machines.
C10.4 Implement artificial intelligent behavior through various methods: mathematical modeling, reinforcement learning, and probabilistic analysis.

College and Career Readiness Standards:

- **Reading**

Key Ideas and details

1.Read closely to determine what the text says explicitly and to make logical inferences from it; cite specific textual evidence when writing or speaking to support conclusions drawn from the text.
2.Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.
3.Analyze how and why individuals, events, or ideas develop and interact over the course of a text.

Craft and Structure

4.Interpret words and phrases as they are used in a text, including determining technical, connotative, and figurative meanings, and analyze how specific word choices shape meaning or tone.
5.Analyze the structure of texts, including how specific sentences, paragraphs, and larger portions of the text (e.g., a section, chapter, scene, or stanza) relate to each other and the whole.
6.Assess how point of view or purpose shapes the content and style of a text.

Integration of Knowledge and Ideas

7.Integrate and evaluate content presented in diverse formats and media, including visually and quantitatively, as well as in words.
8.Delineate and evaluate the argument and specific claims in a text, including the validity of the reasoning as well as the relevance and sufficiency of the evidence.
9.Analyze how two or more texts address similar themes or topics in order to build knowledge or to compare the approaches the authors take.

Range of reading and Level of text Complexity

10.Read and comprehend complex literary and informational texts independently and proficiently.

- **Writing**

Text types and Purposes

1.Write arguments to support claims in an analysis of substantive topics or texts using valid reasoning and relevant and sufficient evidence.
2.Write informative/explanatory texts to examine and convey complex ideas and information clearly and accurately through the effective selection, organization, and analysis of content.
3.Write narratives to develop real or imagined experiences or events using effective technique, well-chosen details and well-structured event sequences.

Production and distribution of Writing

4.Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
5.Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.
6.Use technology, including the Internet, to produce and publish writing and to interact and collaborate with others.
Research to Build and Present Knowledge
7.Conduct short as well as more sustained research projects based on focused questions, demonstrating understanding of the subject under investigation.
8.Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.
9.Draw evidence from literary or informational texts to support analysis, reflection, and research.
Range of Writing
10.Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audience

20. Course Outline

Unit 1: Introduction to html, JavaScript and Computer Programming. In this unit, we will discuss the history of computer programming referencing the many of the names below. We will also be watching excerpts from the five part series: *Crockford on JavaScript* as well as Douglas Engelbart's - *The Mother Of All Demos*. This unit will serve as a basis for the students first research paper.
-Charles Babbage and Ada Lovelace
i) Difference Engine
-History of html and JavaScript
i) Sir Timothy Berners Lee
ii) Douglas Crockford

-Other Pioneers of Programming
i) Bill Gates./Paul Allen
ii) Steve Jobs/ Steve Wozniak
iii) Nolan Bushnell/Allan Alcorn
iii) Ralph Baer
iv) Douglas Engelbart
v) William Higinbotham

Unit 2: Introduction to html Objects. In this unit the students will learn about the basic types of html objects that will be frequently used in writing JavaScript. The students will learn about many of the html tags that are used the most and how they can be incorporated into problem solving. Resources to be used for this chapter will be www.w3schools.com, Chapter 1 Crockford and Chapter 1 Flanagan.
-Divisions (div)
-Input types (text boxes, checkboxes, buttons, etc...)
-Other types (images, links, tables, etc...)
i) Placing objects in a browser using html
ii) Setting attributes of objects (position, background, width, height, etc...)
-Utilizing resources (books, websites et al.) to learn about different objects

Unit 3: Designing a personal webpage using html objects and a cascading style sheet. Students will use the html objects learned about in unit 2 to design a webpage using CSS for positioning, size attributes, colors etc... Resources: www.w3schools.com, Chapter 1 Crockford and Chapter 1 Flanagan.
-CSS (Cascading Style Sheet)
-Object Positioning (absolute, fixed, relative, etc...)
-Setting Object Attributes (id, name, title, etc...)

Unit 4: This unit will focus on an introduction to JavaScript and other computing languages. The focus will be on the common data types, reserved words, mathematical operations and commands that are used across multiple programming languages. The students will be shown variable types not only in JavaScript but in other languages as well. Resources: Chapter 2 Crockford, www.stackoverflow.com
-Data Types (Numbers, Strings, Booleans)
-Declaration of Data Types
-Comparing Data Types and Values
-Variables (Local versus Global)
-Mathematical Operations (Basic operations, Math.operations())
- Arrays and Objects
-Reserved Words (var, Array, Rnd, etc...)
-Reserved Function (toString(), length(), splice(), etc...)
-Commenting in JavaScript and html

Unit 5: This unit will introduce the students to creating a function and the structure of a function. Projects 2 and 3 will be introduced at this point. The students will understand how to create, write, and implement a function to come up with a solution to projects 2 and 3. Students will need to learn how to make function calls from both html objects and other JavaScript functions. Resources: Chapter 4 Crockford, Chapter 8 Flanagan, www.stackoverflow.com, www.codeacademy.com
-Functions vs. Procedures
-Understanding Function Structure
-Global Functions versus Local Functions
-Function Calls
-Referencing html Objects Using JavaScript (id, class, name, etc...)
-Passing Parameters to a Function
-Returning Function Values

Unit 6: This unit will introduce students to many of the procedures that are used across all programming languages. Students will understand how to implement (for example) *if* statements and the structure of the *if* statement for JavaScript and other programming languages. Projects 4 and 5 will be introduced during this unit. Resources: Chapter 5 Flanagan, Chapter 8 Crockford, www.stackoverflow.com, www.codeacademy.com
-If, Then, Else(if) Statements (Ternary Operator)
-For Loops
-While Loops
-Case (Switch) Statements
-Message Boxes
-Debugging and Debuggers

Unit 7: This unit is a continuation of Unit 6 and will continue working with functions and how they relate to html object events. This unit will continue with both projects 4 and 5 which were introduced in unit 6. The students will become more accustomed to creating functions that can b used to solve the problems of projects 4 and 5. Resources: Chapter 17 Flanagan, Chapter 4 Crockford, www.stackoverflow.com, www.codeacademy.com
-Utilizing Different Events (onchange, onmouseover, onclick, etc...)
-Using Events to Call JavaScript Functions
-Using Events to Get and Set Values

Unit 8: Students will learn about dynamic html as opposed to static html using JavaScript. The focus of this unit is to learn how to create different html objects at runtime and assign attributes, style and events using JavaScript. Resources: Chapter 7 Crockford, Chapter 15 Flanagan, www.stackoverflow.com
-Using JavaScript to Create Objects
-Setting Attributes of Objects
-Placing Objects
-Understanding Parent/Child References

Unit 9: This unit will introduce html5 element to be used in projects 5, 6 and 7. The students are to incorporate the new elements into the last four projects. Students will also be informed as to the importance of keeping up with new versions of any computer language as to assist them in becoming dynamic coders. Resources: Chapter 22 Flanagan, www.w3schools.com
-Using the Canvas Element to Create Animation
-Video and Audio Tags for Media Playback
-New Form Controls (Calendar, Email, Search)

Unit 10: The students will be introduced to other languages that are necessary to become a successful programmer. This unit will help with the second research paper of the year. Students will learn in order to be competitive in today's computing environment, that they need to be fluent in more than one or two languages. The languages mentioned below are some of the more popular languages used in the computer world today and would serve them well to learn. Resources: www.codeacademy.com, www.stackoverflow.com
-PHP
-SQL
-C++
-Visual Basic
-Python
-AJAX
-Java

21. Instructional Materials:
Board approved required text

Crockford, D. (2008). *JavaScript: The good parts*. Beijing: O'Reilly.

Supplementary materials

Flanagan, D. (2006). *JavaScript: The definitive guide*. Sebastopol, CA: O'Reilly.

www.w3schools.com
www.stackoverflow.com

www.tizag.com

www.youtube.com

www.codeacademy.com

jsfiddle.net

22. **Instructional Methods and/or Strategies** :Key Assignments

The instructional methods and strategies used for this class will be direct instruction, project-based learning and research and demonstration. It is important for the student to understand that programming languages are not to be memorized but to be utilized. The direct instruction portion of the class will not only demonstrate how a student can find a structure, object or example in regards to the program that they are writing, but how it is not (with all of the resources available today) necessary to memorize all aspects of the language that he/she is coding in.

Since the number of languages continues to grow, and the syntax from one language to the next can vary little or greatly, then it is imperative for the student to be able to research the syntax, format or structure of the commands in the language that they are currently working in.

It is also important that the instructor not try to deviate the student off of their approach to problem solving. The instructor must be able to parse through code and give guidelines as to what can be done to help with any bugs that the student's program may be possess.

Project 1 (weeks 3 and 4) - Create a basic webpage using html and css. The web page must consist of at least three divs. One div must contain links to different web pages that must open in either a new tab or window. The other two divs are for a bio about the designer and images. As designing is subjective, credit will only be lost if the main criteria as mentioned above are not met or the divs overlap.

Project 2 (weeks 7 and 8) - Using html, css and JavaScript to create a basic calculator where the user will have the ability to input two numbers into two different text boxes, and the output will be the sum, difference, product or quotient of the numbers on a button click event. The choice will be made through four different radio buttons that give the user the choice of the mathematical operation.

Project 3 (weeks 11 and 12) - Creating a picture slide show on a click event. An array of pictures will be used to scroll through on a button left or button right click. The page will contain at least two divs and any other html object needed to complete the page. One div is to be used for the slide show and another for a description of the picture that must change as the user clicks to change the photo. If the slide show comes to an end, the user must be taken back to the beginning so as not to force the user to click back to the start of the show.

Project 4 (weeks 14 and 15) - Creating a basic chat interface where two users input messages that are then displayed in a fashion similar to that of a mobile device. There must be a clear way to decipher (by name, color or some other method) who sent which message. Message displays will be created dynamically using JavaScript as opposed to static html objects.

Project 5 (weeks 18, 19 and 20) - Creating a one player racquetball game. Students will use the html5 canvas element along with the setTimeout or setInterval functions to create a simulated ball moving around the canvas element. The game must have the ability to keep score and have adjustments (i.e. faster ball, shrinking paddle) the closer the user gets to 'winning'. The students will use key events to move the paddle back and forth and come up with an algorithm to detect a collision between the ball and the paddle.

Project 6 (weeks 22, 23 and 24) - A continuation of project 5. Recreating Al Alcorn's (and Willy Higinbotham's 'Tennis for Two') Pong. Students will add a second paddle to project 4 and give the program the ability to allow for two players.

Project 7 (weeks 27, 28 and 29) - Creating a 'Simon' game. Students will create a game similar to the classic 'Simon'. The game will use a rectangular grid which will be a function of the level of difficulty chosen by the player. There must be a minimum of three levels.

Project 8 ((Final Project) weeks 32 through the end of the year) - Students will choose their own final project. The project must be interactive and is to be based on the Brain Age game by Nintendo. The student will create an activity to help the user with mathematics, memory, grammar or any other problem solving type issue that requires the user to think through the solution. The final project must be agreed upon by both the students and instructor.

Research Paper 1 (Semester 1): This paper will consist of a brief history of computer programming up to and including the Assembly language. The paper must be 2 1/2 - 3 pages long and must include references to programming pioneers such as Charles Babbage, William Higinbotham and Al Alcorn. The paper must talk about a minimum of three computer pioneers but is also not limited to only three. Please discuss the contributions of the three (or more) that are chosen to modern day computer programming and technology in general.

Research Paper 2 (Semester 2): This paper will discuss the modern era of computer programming post the Assembly language. The paper must be 2 1/2 - 3 pages long and must include references to programming as it has grown since the advent of Assembly. The paper must talk about a minimum of three modern day pioneers such as Bill Gates, Douglas Crockford or Steve Wozniak when it comes to how coding is done today. Please discuss the contributions of the three (or more) that are chosen to modern day computer programming and technology in general. Please conclude your paper with a comparison of pre and post Assembly language and which you prefer.

23. Assessment Methods and/or Tools

The course is mostly project-based, and student projects will be assessed on the following:

Functionality
Style and Syntax
Design
Creativity

Students will have short Unit Quizzes for each unit, to assess their knowledge of syntax, best coding practices, and code interpretation and debugging.

24. Grading Policy

Grading will follow the San Leandro Unified School District's policy. It is expected that assignments will be turned in on time. Late assignments will be accepted if student has communicated to teacher the reasons and actions taken to complete the work in a timely manner.

25. Context for Course:

Software and Systems Development Pathway: ICT

Students in the Software and Systems Development pathway prepare for careers related to computer science that involve the design, development, implementation, maintenance, and management of systems that rely on software programs to satisfy the operational needs of modern business organizations. Persons with expertise in systems development and programming are critical to support operations like electronic commerce, medical records management, retail sales and inventory management, digital entertainment, and use of energy.