SACRAMENTO CITY UNIFIED SCHOOL DISTRICT
BOARD OF EDUCATION

Agenda Item 10.1g

Meeting Date: June 16, 2016

Subject: Approve Revised Course of Study for Computer Science – MXS211 and MXS212

☐ Information Item Only
☒ Approval on Consent Agenda
☐ Conference (for discussion only)
☐ Conference/First Reading (Action Anticipated: ______________)
☐ Conference/Action
☐ Action
☐ Public Hearing

Division: Curriculum and Instruction

Recommendation: Approve the *revised* course of study for "Computer Science"

Background/Rationale: Computer Science provides an introduction to the analysis, design, and implementation of software solutions to simple problems: developing standard algorithms for performing a bubble sort, a linear search of an array, and for data validation. Other topics covered include converting numbers between numbering systems, binary arithmetic including two's complement subtraction, console and file input/output, and functions. Integrated Math I or Algebra I is a required prerequisite for this course. It is a semester-long UC a-g approved course.

The course is designed to parallel the first year computer programming course at the collegiate level. Articulation agreements with local California State Universities and California Community Colleges are highly recommended for any high school teaching this course. If an articulation agreement is in place, students completing this course have the opportunity to receive college credit units.

Financial Considerations: None

LCAP Goal(s): College and Career Ready Students

Documents Attached:
1. Course of Study for "Computer Science"
2. Instructional Materials Adoption

Estimated Time of Presentation: N/A
Submitted by: Matt Turkie, Interim Assistant Superintendent of Curriculum and Instruction
Approved by: José L Banda, Superintendent

COURSE OF STUDY

FOR

Computer Science/Programming 1-2
UCOP.edu: Elective (G)

Segment	High School
Length of Course	One Year
Developed by	*Huang, Jerry (SCUSD)* *Dixon, Michael (Los Rios-SCC)* *Dagler, Clayton (SCUSD)*
First Edition	Spring, 2016

SACRAMENTO CITY UNIFIED SCHOOL DISTRICT

BOARD OF EDUCATION APPROVED ON:

Board of Education

Administration

Committee

Huang, Jerry (SCUSD)
Dixon, Michael (Los Rios-SCC)
Dagler, Clayton (SCUSD)
Pellerin, Jill (SCUSD)
Nick Freathy, SCUSD

“The Sacramento City Unified School District is committed in all of its activities, policies, programs, and procedures to provide equal opportunity for all to avoid discrimination against any person regardless of ethnicity, gender, religion, national origin, disability, marital status, or age.”

SECTION ONE — GENERAL INFORMATION

TABLE OF CONTENTS

SECTION TWO — COURSE UNITS

Computer Science/Programming 1-2

SECTION ONE — GENERAL INFORMATION

COURSE DESCRIPTION

This course provides an introduction to the analysis, design, and implementation of software solutions to simple problems: developing standard algorithms for performing a bubble sort, a linear search of an array, and for data validation. Other topics covered include converting numbers between numbering systems, binary arithmetic including two's complement subtraction, console and file input/output, and functions.

Prerequisites	
Integrated Math I or Algebra I	Required

RATIONALE

Computer programming is the essential skill of developing software applications, controlling robots, and designing the world around us today. More so than ever, information and computer technologies lead our economy and industries.

In addition, this course is designed to parallel the first year computer programming course at the collegiate level. Articulation agreements with local California State Universities and California Community Colleges are highly recommended for any high school teaching this course. If an articulation agreement is in place, students completing this course have the opportunity to receive college credit units. (Most colleges and universities require students to have a grade/mark of an A or B and sometimes other requirements set by the college or university professor).

COURSE GOALS (COURSE OUTCOMES)

Upon completion of this course, the student will be able to:

- analyze simple problems to fully understand the scope and desired outcome.
- create algorithms for solving simple problems demonstrating use of variables, constants, and the proper use of control structures such as simple sequence, selection, and iteration.
- document algorithms using structure charts, pseudocode, and flowcharts.
- implement, test, and debug a program, based on a documented algorithm, which uses each of the following fundamental programming constructs: basic computation, simple console and file input/output, standard conditional and iterative structures, and the definition of functions.
- demonstrate understanding of high to low level language translation.
- create a test data document that states the testing criteria to thoroughly test a program for completeness and accuracy.
- evaluate the completeness and accuracy of a program in accordance with test data previously created.

- demonstrate the mechanics of parameter passing.
- demonstrate understanding of bubble sort, linear searches of an array, and data validation algorithms.

COURSE CONTENT

Topics covered begin with the general model of computing, variables, data types, expressions, inputs and outputs, and arrays. Then students will learn top-down design, iterative design, pseudocoding, testing, and critiquing a design. From there, students will learn how to use logical operators and control structures (if, while, for, etc.) provided by the programming language. In addition, students will develop an understanding of sorting algorithms, binary numbers, American Standard Code for Information Interchange (ASCII), and UNICODE.

As a means of instruction, this course will utilize programming languages and/or integrated development environments (IDE). As an end, the students should create applications for a target device (personal computer, mobile device, etc.)

COURSE STANDARDS

Common Core Math Standards

CCSS.Math.Practice.MP1
Make sense of problems and persevere in solving them.
CCSS.Math.Practice.MP3
Construct viable arguments and critique the reasoning of others.
CCSS.Math.Practice.MP4
Model with mathematics.
CCSS.Math.Practice.MP5
Use tools strategically.
CCSS.Math.Practice.MP7
Look for and make use of structure.
CCSS.Math.Content.6.EE.A.2
Write, read, and evaluate expressions in which letters stand for numbers.
CCSS.Math.Content.6.NS.B.3
Fluently add, subtract, multiply, and divide multi-digit decimals using the standard algorithm for each operation.
CCSS.Math.Content.6.NS.C.6
Understand a rational number as a point on the number line. Extend number line diagrams and coordinate axes familiar from previous grades to represent points on the line and in the plane with negative number coordinates.
CCSS.Math.Content.6.NS.C.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.
CCSS.Math.Content.6.EE.B.3

Solve multi-step real-life and mathematical problems posed with positive and negative rational numbers in any form (whole numbers, fractions, and decimals), using tools strategically. Apply properties of operations to calculate with numbers in any form; convert between forms as appropriate; and assess the reasonableness of answers using mental computation and estimation strategies.

CCSS.Math.Content.6.EE.B.4

Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

CCSS.Math.Content.HSA-CED.A.2

Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales.

Common Core English Language Arts Standards

CCSS.ELA-Literacy.CCRA.SL.1

Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.

CCSS.ELA-Literacy.CCRA.SL.2

Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively, and orally.

CCSS.ELA-Literacy.CCRA.SL.3

Evaluate a speaker's point of view, reasoning, and use of evidence and rhetoric.

CCSS.ELA-Literacy.CCRA.SL.4

Present information, findings, and supporting evidence such that listeners can follow the line of reasoning and the organization, development, and style are appropriate to task, purpose, and audience.

CCSS.ELA-Literacy.CCRA.SL.5

Make strategic use of digital media and visual displays of data to express information and enhance understanding of presentations.

CCSS.ELA-Literacy.CCRA.SL.6

Adapt speech to a variety of contexts and communicative tasks, demonstrating command of formal English when indicated or appropriate.

CCSS.ELA-Literacy.CCRA.L.6

Acquire and use accurately a range of general academic and domain-specific words and phrases sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when encountering an unknown term important to comprehension or expression.

CCSS.ELA-Literacy.RST.11-12.3

Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.

CCSS.ELA-Literacy.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*.

CCSS.ELA-Literacy.RST.11-12.9

Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.

CTE / Information Communication Technology Standards

2.0 Communications

Acquire and accurately use Information and Communication Technologies sector terminology and protocols at the career and college readiness level for communicating effectively in oral, written, and multimedia formats. (Direct alignment with LS 9-10, 11-12.6)

3.0 Career Planning and Management

Integrate multiple sources of career information from diverse formats to make informed career decisions, solve problems, and manage personal career plans. (Direct alignment with SLS 11-12.2)

4.0 Technology

Use existing and emerging technology, to investigate, research, and produce products and services, including new information, as required in the Information and Communication Technologies sector workplace environment. (Direct alignment with WS 11-12.6)

5.0 Problem Solving and Critical Thinking

Conduct short, as well as more sustained, research to create alternative solutions to answer a question or solve a problem unique to the Information and Communication Technologies sector using critical and creative thinking, logical reasoning, analysis, inquiry, and problem-solving techniques. (Direct alignment with WS 11-12.7)

9.0 Leadership and Teamwork

Work with peers to promote divergent and creative perspectives, effective leadership, group dynamics, team and individual decision making, benefits of workforce diversity, and conflict resolution such as those practiced in the Future Business Leaders of America and SkillsUSA career technical student organization. (Direct alignment with SLS 11-12.1b)

10.0 Technical Knowledge and Skills

Apply essential technical knowledge and skills common to all pathways in the Information and Communication Technologies sector, following procedures when carrying out experiments or performing technical tasks. (Direct alignment with WS 11-12.6)

11.0 Demonstration and Application

Demonstrate and apply the knowledge and skills contained in the Information and Communication Technologies anchor standards, pathway standards, and performance indicators in classroom, laboratory, and workplace settings, and through career technical student organizations such as Future Business Leaders of America and SkillsUSA.

INSTRUCTIONAL MATERIALS

THE FOLLOWING IS A REPRESENTATIVE LIST OF TEXTBOOKS AS THE WORLD OF COMPUTER SCIENCE AND PROGRAMMING CONTINUES TO CHANGE:

IDE Software: Visual Studios

Developer: Microsoft Corporation

https://www.visualstudio.com

Language: C#

Developer: Microsoft Corporation

https://www.visualstudio.com (contained with Visual Studios)

Language: Python

Developer: Python Software Foundation

https://www.python.org (integrates with Visual Studios)

IDE Software: Xcode

Developer: Apple Inc.

https://developer.apple.com/xcode

Language: Swift

Developer: Apple Inc.

https://developer.apple.com/swift

IDE Software: Code::Blocks

Developer: The Code::Blocks team

http://codeblocks.org

Language: C++

Developer: Bjarne Stroustrup

http://codeblocks.org (contained with Code::Blocks)

SUGGESTED AVERAGE TIME FOR COVERING MAJOR UNITS

Unit 1	General Model of Computing		5 hours
Unit 2	Variables, Expressions, Input and Output		10 hours
Unit 3	Intro to Program Analysis and Design		20 hours
Unit 4	Intro to Logical Operators		20 hours
Unit 5	Iteration Techniques (Loops) and Constants		20 hours
Unit 6	Top-down design with Functions		20 hours
Unit 7	One-Dimensional Arrays		10 hours
Unit 8	File Input/Output		10 hours
Unit 9	Linear Search and Sorting Algorithms		15 hours
Unit 10	Data and Program Representation and Storage		10 hours
Unit 11	Suggested Applications		40 hours
		TOTAL	180 hours

RECOMMENDED TEACHER AND STUDENT RESOURCES

THE FOLLOWING IS A REPRESENTATIVE LIST OF TEXTBOOKS AS THE WORLD OF COMPUTER SCIENCE AND PROGRAMMING CONTINUES TO CHANGE:

Stewart Venit (2006). *Concise Prelude to Programming: Concepts and Design* (3rd ed.). Addison Wesley Higher Education. [ISBN: 0321482662]

Thad Crews and Chip Murphy (2009). *A Guide to Working Visual Logic*. Course Technology, Cengage Learning. [ISBN: 03246011901]

SECTION TWO — COURSE UNITS

Unit 1: General Model of Computing

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP7

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.4

CTE / Information Communication Technology Standards
2.0 Communications

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C1.1 Identifying 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 and existing software applications.
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.

Unit Description

In this unit students see the relationship between hardware, software, people, and processes. This includes the types of hardware, software, and how they interact with people. The students will also learn the systems development life cycle (SDLC).

Instructional Objectives

Students will be able to...

- Differentiate hardware and software components of a computer.
- Articulate the purpose of hardware and software elements and their functions in a computer system.
- Diagram the input and output interacts between people and computers.
- Use computers processes to solve problems.
- Have a basic understanding of the system development life cycle (SDLC).

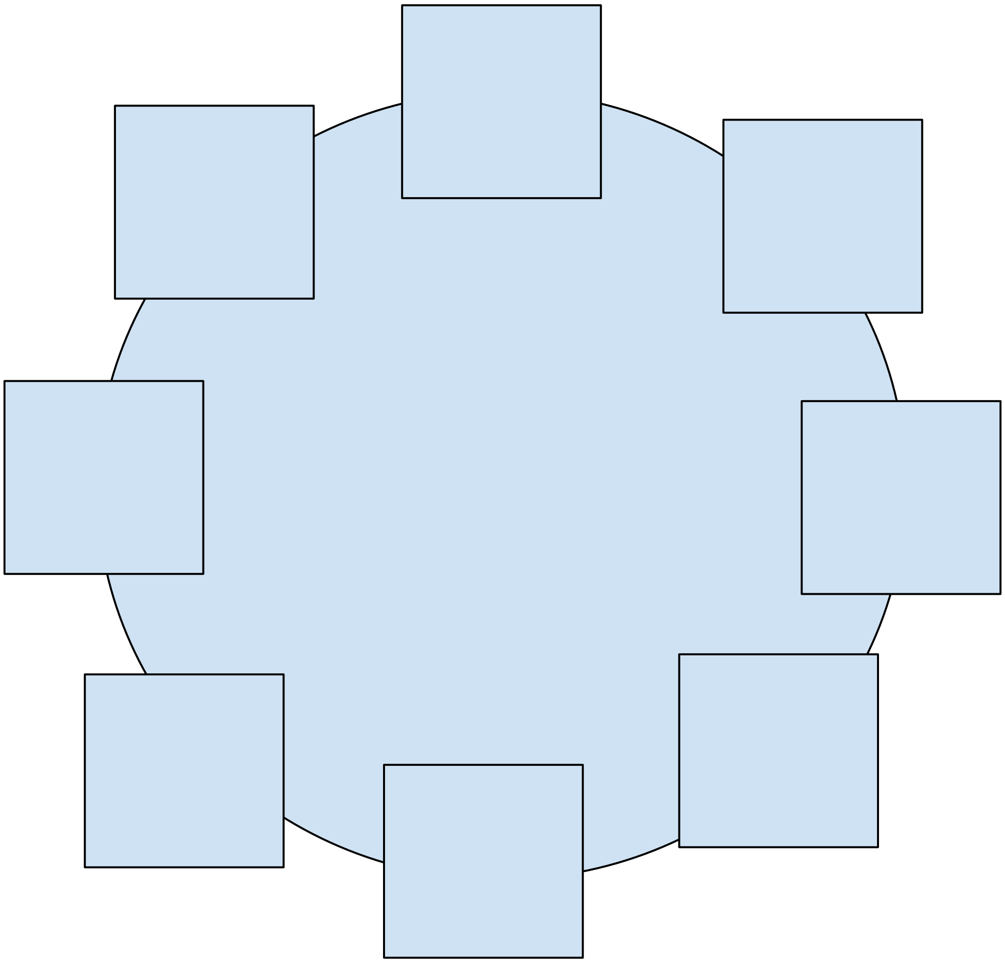
Suggested Activities

1. The students first need to list all of the components of their smart phone and then organize these components into categories.
2. Have the students talk about the advantages and disadvantages of different types of hardware memory (CACHE, RAM, SDD, and HD)
3. Entire class attempts to list all the input and output devices found in the classroom. Discuss the similarity of all the input devices and output devices when relating and interacting with a human.
4. Students make comic strip that involves a person interacting with at least one computer input and one computer output. (*https://www.bitstrips.com/create/comic/*)

Suggested Assessments

Teacher made test and quizzes (below are some assessment ideas).

Fill in the blank of a SDLC diagram:



Students list in a T-Table examples of hardware and software:

Hardware examples	Software examples

Unit 2: Variables, Expressions, Input and Output

Content Standards

Common Core Math Standards
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
3.0 Career Planning and Management
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C2.0 Define and analyze systems and software requirements.
C3.0 Create effective interfaces between humans and technology.
C4.0 Develop software using programming languages.

Unit Description

In a math class, variables are only used to store numbers, where a variable in programming can also be used to store letters, words, boolean values and more. In programming, a variable is a named location that stores a value. Expressions are used in programming to process calculations which solve simple application. A computer program gets values from the use with input and shows that answer using output. In this unit students will begin their first computer programs and will learn to debug code.

Instructional Objectives

Students are able to…

- understand the similarity and differences between a math and computer language variable.

- understand the similarity and differences between a math and computer language expressions.
- work with variables in a given programming language syntax
    - declaring variables
    - use variables in an expression
    - assign values to a variable
    - store user input in a variable
    - use variables values to create an output (print to a screen, play tones, etc.)
- debug code by understanding error messages displayed in the IDE.

Suggested Activities

1. Students write small program to evaluate mathematical equations/expressions:
    - Students write programs that convert from fahrenheit to celsius.
    - Find the distance a car traveled at a fixed rate and given time.
    - Find the slope of a line through two points.
    - Solve a polynomial of degree 2 using the quadratic formula.
2. Students are given small programs with syntax errors and are faced with getting the program to compile:
    - Missing semi-colon (language specific)
    - Mis-matched parentheses
    - Undeclared variables
    - Using spaces inside variable names.
3. Write the classic ‘Hello World’ program for the given programming language

Suggested Assessments

Teacher made test and quizzes (below are some assessment ideas).

- Printed out code with syntax errors and students identify them.
- Students write a program that converts inches to centimeters.
- Students input the coefficients of a quadratic equation and then outputs the x-value of is maximum or minimum value.
- Students write a program that reads in a user’s name prints out a greeting message.

Unit 3: Intro to Program Analysis and Design

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C3.0 Create effective interfaces between humans and technology.
C4.0 Develop software using programming languages.
C5.0 Test, debug, and improve software development work.

Unit Description

Program Analysis and Design: problem analysis and design of a structured solution using top-down design and sequence, selection, and iteration techniques; documentation of high-level and detailed designs using structure charts, flowcharts, and pseudocode; identification of test criteria to be applied during testing. Students will complete a project that utilizes these techniques after following the development life cycle.

Instructional Objectives

Students will be able to…

- Plan and design a project with a top-down approach
- Write pseudocode as a part of top-down design
- Use flowcharts to segment the phases of a computer program
- Practice sequence, selection, and iteration techniques

Suggested Activities

Students will write a solution to the following problems using structured charts, flowcharts, and/or pseudocode. Once the teacher approves the design the students are to write the code that solves the problem:

- The manager of the Lakeview Hotel wants a program that calculates and displays a guest’s total bill. Each guest pays a room charge of $170.00 each night. The guests are also charged $17.50 for each for each time they use room service. Write a program which solves this problem.
- The accountant at Printing Haven wants a program that will help her prepare a customer’s bill. Each page that is printed costs 10₵ and each envelop costs 23₵. There is also a tax rate of 5%. The program needs to print out the subtotal, tax and total.
- Shannon opened a savings account in 2010 and deposited $2500.00 as principal. The account earns 12% interest, compounded annually. Write a program that reads in what year it is and then finds the balance for that year.

Suggested Assessments

- Students present a collaborative design project
    - Includes initial brainstorming and high-level planning evidence
    - Includes flowcharts and/or structured charts
    - Includes critiques and iterative design changes

Includes debugging challenges and logical design errors faced during project

Unit 4 : Intro to Logical Operators

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C3.0 Create effective interfaces between humans and technology.
C4.0 Develop software using programming languages.
C5.0 Test, debug, and improve software development work.

Unit Description

This unit introduces logical statements, commonly known as ‘if-statements’ in most programming languages. Logical statements will be used to control which statements in computer code is or is not executed based on testing of logical expressions using operators such as ‘less than’, ‘less than or equal to’, ‘equal to, etc. Students will complete a project that utilizes these techniques after following the development life cycle.

Instructional Objectives

Students will be able to…

- Write logical statements to control the flow of a computer program.
- Use mathematical comparators like >=, <, ==, etc. in a programming language to dynamically compare variables and values.

Suggested Activities

Students will write a solution to the following problems using structured charts, flowcharts, and/or pseudocode. Once the teacher approves the design the students are to write the code that solves the problem:

- A certain city classifies a pollution index less than 35 as “Pleasant”, 35 through 60 as “Unpleasant” and above 60 as “Hazardous”. Write a program pollution.cpp that reads in a pollution index and then displays the appropriate classification.
- Write a program that reads in a letter grade (A, B, C, D, or F) and uses a switch statement to print out:

You are the best!	For ‘A’ input
You did a good job!	For ‘B’ input
Your hard work paid off.	For ‘C’ input
Done.	For ‘D’ input
Do better next time	For ‘F’ input
What grade did you get?	For all other inputs

- Write a program that reads in the height and radius of a cylinder and then ask the user if he/she wants to find its surface area, lateral surface area or its volume. The program is then to find what the user wants to calculate and then outputs it.
- Write a program that determines if a students can give blood or not. A person can give blood if he/she is 16 years or older and weighs over 110 pounds. If the person is too young it needs to print out: “sorry you are too young”, if the person is old enough but does not weigh enough, it needs to print out: “I am sorry but you just do not weigh enough”, otherwise it needs so say: “look away you are going to get stabbed and thanks for saving lives”.

Suggested Assessments

- Students are given a flowchart diagram and a given value and must determine the output.
- Students present a collaborative design project
  - Includes initial brainstorming and high-level planning evidence
  - Includes flowcharts and/or structured charts
  - Includes critiques and iterative design changes

- ○ Includes debugging challenges and logical design errors faced during project

Includes evidence of how logical statements are used to control which statements in computer code are executed and which statements are not.

Unit 5: Iteration Techniques (Loops) and Constants

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C3.0 Create effective interfaces between humans and technology.
C4.0 Develop software using programming languages.
C5.0 Test, debug, and improve software development work.

Unit Description

Topics presented in this unit include an introduction to the use of loops such as ‘for loops’ and ‘while loops’ to handle repeating code. Also this unit also presents the use of constants to make code clearer and easily to maintain. Students will complete a project that utilize these techniques after following the development life cycle.

Instructional Objectives

Students will be able to…

- Use iterative techniques to reuse code.
- Use constants to define key values.

Suggested Activities

Students will write a solution to the following problems using structured charts, flowcharts, and/or pseudocode. Once the teacher approves the design the students are to write the code that solves the problem:

- Write a program sequence3.cpp that list the first 20 numbers in the sequence:
  7, 21, 63, 189, ...
- Write a program *ball.cpp* that displays the rebound height of a dropped ball. This ball rebounds to half of its height that it was dropped. The program needs to read in the ball's initial height and then output its height for the first 20 bounces. The input/output should look like:

  ```
  Finding the rebound height of a dropped ball.
  Enter the original height (in meters): 15
          Starting height:      15.0000 meters.
          After bounce # 1:      7.5000 meters
          After bounce # 2:      3.7500 meters
          After bounce # 3:      1.8750 meters
          …
  ```
- Write a program *series1.cpp* that finds the sum of the first 50 numbers in the sequence:
  80000, 64000, 51200, 40960, 32768, ...
- Write a program the reads in a set of numbers of unknown size. Once all the numbers are inputted, the program needs to output the average of the numbers. Once a number is inputted, the program needs to ask the user if he/she would like to input another number.

Suggested Assessments

- Students present a collaborative design project
  - Includes initial brainstorming and high-level planning evidence
  - Includes flowcharts and/or structured charts
  - Includes critiques and iterative design changes
  - Includes debugging challenges and logical design errors faced during project
  - Includes evidence of how iterative Techniques (loops) are used to minimize code.

Includes evidence of how constants are used to make code clearer and easier to maintain.

Unit 6: Top-down design with Functions

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C3.0 Create effective interfaces between humans and technology.
C4.0 Develop software using programming languages.
C5.0 Test, debug, and improve software development work.

Unit Description

This unit introduces the creation and use of functions as an approach to programmatic control of code sequence. Students will first learn how to pass values into a function and how they return values. While doing this, the students will understand the difference between passing a variable by value and passing a variable by reference. Once this is mastered, the students will understand the scope of a variable. Students will complete a project that utilize functions after following the development life cycle.

Instructional Objectives

Students will be able to…

- Create variables in a program with a clear understanding of variable scope.
- Create functions based on a top-down design view of the entire functionality of the program.
- Calculate and return data with the the proper datatype and value.

Suggested Activities

Students will write their own mathematical operation as a function that inputs two values and returns its answer.

Suggested Assessments

- Students are given flowcharts of the main program and separate flowchart diagrams for each function. Given a value, the student must determine the output.
- Students present a collaborative design project
    - Includes initial brainstorming and high-level planning evidence
    - Includes flowcharts and/or structured charts
    - Includes critiques and iterative design changes
    - Includes debugging challenges and logical design errors faced during project
    - Includes evidence of how their program utilizes functions
    - Includes analysis of the scope of variables in the program
    - Includes discussion if a function uses a copy or the original variable of the caller.

Unit 7 : One-Dimensional Arrays

Content Standards

Common Core Math Standards

CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.NS.B.3
CCSS.Math.Content.6.NS.C.6
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.3
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.SL.6
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Unit Description

An array stores a sequence of values that are all of the same type. An array can be thought of as a single variable what stores multiple values. Each of the values in an array has an indexed number attached to it. This index number can be use to quickly access the value. One good way to think of an array is to see it as one row or column in a spreadsheet. In this unit students will use arrays to solve problems that involve a large amount of data of the same type.

Instructional Objectives

Students will be able to…

- Identify when to utilize an array for a program.
- Create and properly utilize one dimensional arrays for a program.
- Use the index number to access (read or modify) elements in a one dimensional array.

Suggested Activities

- Students will create a program with a for loop that will iterate over an one dimensional array. Some examples:
    - Printing out a list of names stored in an array with an entire classroom student roster.
    - Averaging a list of test grades stored in an array with all test grades of a course.
- Students will create a program that selectively accesses elements in an one dimensional array Some examples:

    - Printing out the name of a student sitting at a particular computer number based on an array of all students indexed by computer station number.
    - Select an inventory item (string value) based inventory index number. (Useful in modeling an inventory of items for a game like Minecraft).

Suggested Assessments

- When given a code fragment, the student must determine the array cell indices and values accessed/modified/printed.
- Students present a collaborative design project
    - Includes initial brainstorming and high-level planning evidence
    - Includes flowcharts and/or structured charts
    - Includes critiques and iterative design changes
    - Includes debugging challenges and logical design errors faced during project
    - Includes evidence of how their program utilizes a one dimensional array

Unit 8: File Input/Output

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4
CCSS.Math.Content.HSA-CED.A.2

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.3
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.SL.6
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills

Unit Description

Up to this point the input and output for our programs have been limited to the console. This works well if the programing is only inputting and outputting small quantities of data. Using files to input and output data to and from a program is much more efficient and accurate when dealing with medium to a large amount of data. In this unit, students will write programs that read in data from a file and write programs that also write data to a file.

Instructional Objectives

Students will be able to…

- Identify when to utilize file I/O.
- Programmatically read and write data to files.

Suggested Activities

- Students will write code that allows the user to type their name and then saves the name is a file on a networked drive. Then students will write code that reads from the same network drive file but from a different computer and writes it to the screen.
- Students will write code that can save and read a “game state” like tic-tac-toe.

Suggested Assessments

- Students complete a fill-in-the-blank assessment that is an entire program where the read/write/append code is incomplete.
- Students present a collaborative design project
  - Includes initial brainstorming and high-level planning evidence
  - Includes flowcharts and/or structured charts
  - Includes critiques and iterative design changes
  - Includes debugging challenges and logical design errors faced during project
  - Includes evidence of how their program utilizes file input and output

Unit 9: Linear Search and Sorting Algorithms

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.NS.B.3

CCSS.Math.Content.6.EE.B.4

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.3
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.SL.6
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
3.0 Career Planning and Management
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Unit Description

Linear search is the process of looking through an array one element at a time for a certain value until it is found or reaches the end of the array. In this unit, students will design and write programs that look for values of different data types in an array.

Arranging an array of values in a certain order is called sorting. An array of values can be sorted two different ways: ascending order is when the first element in the array is the smallest "value" and descending order where the first element in the array is the "largest" value.

Instructional Objectives

Students will be able to…

- programmatically implement a search algorithm to find an element in an array.
- programmatically implement a sorting algorithm to arrange elements in an array.

Suggested Activities

- Students will sort a collection of physical items (collectible cards, toys, etc.) based on their determination. Then students will describe how they approached the problem. Then students will compare their approach to common computer program sorting techniques such as bubble sort (insertion sort, etc.)

- Students will perform a linear search of a word in the dictionary. As a class, students will brainstorm better techniques that a computer could implement to make the search faster and more efficient.

Suggested Assessments

- Given the various sorting algorithms as a flowchart, students will write the state of the data for each pass of the sorting algorithm.
- Students write literacy task that recommends different search and/or sorting algorithms based on a programmer's application.
- Students present a collaborative design project
    - Includes initial brainstorming and high-level planning evidence
    - Includes flowcharts and/or structured charts
    - Includes critiques and iterative design changes
    - Includes debugging challenges and logical design errors faced during project
    - Includes evidence of how their program utilizes a sorting algorithm
    - Includes evidence of how their program utilizes a search algorithm

Unit 10: Data and Program Representation and Storage

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3.
CCSS.Math.Practice.MP5
CCSS.Math.Practice.MP7

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.SL.6
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4

CTE / Information Communication Technology Standards
4.0 Technology
5.0 Problem Solving and Critical Thinking

Unit Description

We may not think of it but the number system we use is base 10. This is because we only use 10 different symbols {0, 1, …, 9} to represent any number. Another numbering system called hexadecimal is based 16 because it uses 16 different symbols to represent numbers {0, 1, …, 9, A, B, C, E, F}. For example, the number eleven would be B in hexadecimal. Binary is another number system which only uses two symbols to represent numbers {0 and 1}. Binary is an important number system today because it is how a computer stores data. In this unit, students will convert to and from these different numbering systems.

Instructional Objectives

Students will be able to…

- Convert numbers between different number systems, especially base 10, binary, and hexadecimal.
- Develop a number system for computers.
- Convert a number to the Two's Complement programming number system.
- Evaluate basic mathematical operations in different number systems.
- Encode and decode using symbology standards of: ASCII, EBCDIC, or UNICODE

Suggested Activities

- Students will secretly encode messages using ASCII, EBCDIC, or UNICODE tables
- Students will intentionally create math errors by asking a computer program to calculate a number that has a value greater than the max/min of a given datatype.
- Students will be asked to create their own method of representing negative numbers for a computer that can only store one and zeroes. Then students will compare their approach with Two's Complement.

Suggested Assessments

- Students complete a series of problems that they are required to…
    - convert a number to Two's Complement and vice versa
    - convert numbers between binary, decimal, and hexadecimal
    - add and subtract binary numbers
- Students will write a literacy task where they will compare and contrast the advantages and disadvantages of American Standard Code for Information Interchange (ASCII) vs. Extended Binary Coded Decimal Interchange Code (EBCDIC) vs. Unicode

Unit 11: Suggested Applications

Content Standards

Common Core Math Standards
CCSS.Math.Practice.MP1
CCSS.Math.Practice.MP3
CCSS.Math.Practice.MP4
CCSS.Math.Practice.MP5
Use tools strategically.
CCSS.Math.Practice.MP7
CCSS.Math.Content.6.EE.A.2
CCSS.Math.Content.6.NS.B.3
CCSS.Math.Content.6.NS.C.6
CCSS.Math.Content.6.NS.C.8
CCSS.Math.Content.6.EE.B.3
CCSS.Math.Content.6.EE.B.4
CCSS.Math.Content.HSA-CED.A.2

Common Core English Language Arts Standards
CCSS.ELA-Literacy.CCRA.SL.1
CCSS.ELA-Literacy.CCRA.SL.2
CCSS.ELA-Literacy.CCRA.SL.3
CCSS.ELA-Literacy.CCRA.SL.4
CCSS.ELA-Literacy.CCRA.SL.5
CCSS.ELA-Literacy.CCRA.SL.6
CCSS.ELA-Literacy.CCRA.L.6
CCSS.ELA-Literacy.RST.11-12.3
CCSS.ELA-Literacy.RST.11-12.4
CCSS.ELA-Literacy.RST.11-12.9

CTE / Information Communication Technology Standards
2.0 Communications
3.0 Career Planning and Management
4.0 Technology
5.0 Problem Solving and Critical Thinking
9.0 Leadership and Teamwork
10.0 Technical Knowledge and Skills
11.0 Demonstration and Application

Software and Systems Development Pathway Standards
C1.0 Identify and apply the systems development process.
C2.0 Define and analyze systems and software requirements.
C3.0 Create effective interfaces between humans and technology.
C4.0 Develop software using programming languages.
C5.0 Test, debug, and improve software development work.
C6.0 Integrate a variety of media into development projects.
C7.0 Develop Web and online projects.
C8.0 Develop databases.
C9.0 Develop software for a variety of devices, including robotics.
C10.0 Develop intelligent computing
(depends on implementation)

Unit Description

The activities and assessments listed below are a list of suggestions to theme and approach the goals of the first ten units of this course.

NOTE: The activities and assessments given below are summative because it requires the students to apply the skills they are learned in the class. At the same time, these activities show real life examples on how programming can be used in the real world. The instructor can choose which activity to do based on a class theme and/or resources that are available.

Instructional Objectives

Students will be able to…

- Develop software products using industry standard platforms.
- Target actual hardware products with their own software.
- Solver real-world problems with software design.

Suggested Activities and Assessments

- Literacy Task Assessments
    - Students write a task for a robot to solve.
    - Students write a design proposal for a new application based on field research of surveys, needs, and trends.
- Collaborative Design Projects
    - Create a mobile application for a mobile device (tablet, phone, etc.)
    - Create a game application for a game console (XBOX, PS4, Arduino, etc.)
    - Create an animation (C-STEM-QAnimate, Visual Studios/XNA, etc.)
    - Create a web application (JavaScript, Adobe Flash, HTML5, etc.)
    - Create and program a robot as a tool or problem solver (LEGO Mindstorms, Barobo, etc.)
        - Musical instrument
        - Rubik’s Cube Solver
        - R/C Car
        - Board game companion/competitor

Sacramento
City Unified
School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Title of Course of Study: *Computer Science*

Title of Course for Instructional Materials Adoption: *N/A*

Date Work Completed: *4/15/2016*

Please check one:

____ School use only—Identify school(s):
__X__District-wide use

A. PROCESS/PROCEDURES

_X_ Curriculum Development
___ Instructional Materials Adoption:

List the Name of Course Developers/Writers with Title/Position:

Jerry Huang, Math & Computer Science CTE Teachers @New Tech
Clayton Dagler, Math & Computer Science Teacher @Burbank
Michael Dixon, Computer Science Professor @Sac City College

DESCRIBE COMMITTEE PROCESS AND PROCEDURES USED.

Composition of Committee/ Names of Reviewers:

Nick Freathy, Math Training Specialist
Jill Pellerin, ELA Training Specialist
Roscoe Anh, Math & Computer Science Teacher @CKM
Joseph Stymeist, Interim Director of CCR & former Computer Science Teacher
Dr. Lily Liemthongsamout, Linked Learning Coordinator

Roles and responsibility of the committee/reviewers:

All committee members listed above reviewed the course of study for Computer Science based on their areas of expertise. Reviewers provided feedback to the course developers and revisions were made in the final version.

Meeting dates and purpose of the meetings:

Two days in January – Jerry and Clayton met with Michael Dixon, Professor at Sac City College to work on course of study
Review of course happened via email. No in-person meeting was held.

Describe Input/Feedback from Community and/or Industry Professionals, if any: (school/pathways, teachers, administrators, public, etc.):

Input and feedback received from a professor at Sac City College and the Capital Academies and Pathways Sector Coach and Career Specialists

B. SEQUENCING

Describe the course work sequence which includes this course:

The sequence of courses will be 1) introduction to computer, 2) exploring computer science, 3) computer science, 4) computer programming.

C. ALIGNMENT MATRIX

I. How does this course align to the California Content Standards and/or CA Common Core Standards?

It covers several standards in Common Core Math, Common Core English Language Arts Standards, and CTE / Information Communication Technology Standards

II. If the course is proposed as part of a Linked Learning Pathway or Career Academy, how does it align with the pathway and/or academy outcomes?

It can be the capstone class for an Information Technology pathway.

III. How does the course engage students in academic work that meets the district's definition of academic rigor (academic rigor ="The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.")?

The course is aligned to a college level Computer Programming Class.

IV. Did/will the course meet the requirements for UC a-g approval?

Yes: Elective (G)

V. How does this course satisfy High School Graduation Requirements?

It meets elective credit.

VI. Other Comments:

For Office Use Only---Approvals:

Dates each group approved the completed product:

- Principals:
- Assistant principals of instruction:
- High school curriculum committee:
- Cadre:
- Superintendent Cabinet:

Sacramento
City Unified
School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Reviewers' Feedback & Recommendation

Title of Course of Study:	*Computer Science*
Name and Position/Title of Reviewer 1:	*Roscoe Anh, Math and Computer Science Teacher @CK McClatchy HS*
Name and Position/Title of Reviewer 2:	*Clayton Dagler, Math and Computer Science Teacher @Luther Burbank HS*
Name and Position/Title of Reviewer 3:	*Joseph Stymeist, Interim Director of College and Career Readiness Department (former computer science teacher)*
Date Review Completed:	*4/15/2016*

Components and Criteria of a High Quality and Rigorous Course of Study:

- The course of study is a course within a clear and coherent sequenced of courses.
- The course of study is aligned to the CA Content Standards and/or CA Common Core Standards
- The course of study is aligned to the academy/pathway Program of Study and Student Learning Outcomes.
- The course engages students in academic work that meets the district's definition of Academic Rigor: *The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.*

☑ The course of study meets all criteria of a high quality course of study. We recommend it be approved for implementation beginning in 2016-17.
This course is based on the curriculum from the http://www.exploringcs.org/curriculum, which is most to date curriculum on computer science.

□ The course of study meets most criteria of a high quality course of study. We recommend it be revised to include the following components prior to approval for implementation beginning in 2016-17:

□ The course of study meets some criteria, but is lacking key components of a high quality course of study. We recommend the course be revised to include the following components prior to approval for Pilot Status in 2016-17.

□ The course of study lacks all or most criteria of a high quality course of study. We recommend the course be rewritten with guidance and support from the district curriculum development staff.

Signatures
Reviewer 1: Reviewer 2: Reviewer 3:

If more space is needed for comments/feedback, attach separate page.

Sacramento
City Unified
School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Title of Course of Study: *Computer Science*

Title of Course for Instructional Materials Adoption: *N/A*

Please check one:

___School Used Only – **Identify school(s):**
X__District-Wide Use

Course of Study Reviewer & Title/Position: *Roscoe Anh, Math & Computer Science Teacher @CKM*

Date Review Completed: *4/15/2016*

A. PROCESS/PROCEDURES

X__Curriculum Development
___Instructional Materials Adoption:

B. ALIGNMENT MATRIX

I. How does this course align to the California Content Standards and/or CA Common Core Standards?

It is well aligned to standards.

II. If the course is proposed as part of a Linked Learning Pathway or Career Academy, how does it align with the pathway and/or academy outcomes?

This course will prepare students to enter into an entry-level programming position.

III. How does the course engage students in academic work that meets the district's definition of academic rigor (academic rigor ="The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.")?

Computer Science requires critical thinking and problem solving skills and forces students to think logically and methodically. Looping, recursion, abstraction, and conditionals are some examples of applied higher-level thinking skills.

IV. Other Comments:

For Office Use Only---Approvals:

Dates each group approved the completed product:

- Principals:
- Assistant principals of instruction:
- High school curriculum committee:
- Cadre:
- Superintendent Cabinet:

Sacramento
City Unified
School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Reviewers' Feedback & Recommendation

Title of Course of Study:	*Computer Science*
Name and Position/Title of Reviewer 1:	*Roscoe Anh, Math and Computer Science Teacher @CK McClatchy HS*
Name and Position/Title of Reviewer 2:	*Clayton Dagler, Math and Computer Science Teacher @Luther Burbank HS*
Name and Position/Title of Reviewer 3:	*Joseph Stymeist, Interim Director of College and Career Readiness Department (former computer science teacher)*
Date Review Completed:	*4/15/2016*

Components and Criteria of a High Quality and Rigorous Course of Study:

- The course of study is a course within a clear and coherent sequenced of courses.
- The course of study is aligned to the CA Content Standards and/or CA Common Core Standards
- The course of study is aligned to the academy/pathway Program of Study and Student Learning Outcomes.
- The course engages students in academic work that meets the district's definition of Academic Rigor: *The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.*

☑ The course of study meets all criteria of a high quality course of study. We recommend it be approved for implementation beginning in 2016-17.
This course is based on the curriculum from the http://www.exploringcs.org/curriculum, which is most to date curriculum on computer science.

☐ The course of study meets most criteria of a high quality course of study. We recommend it be revised to include the following components prior to approval for implementation beginning in 2016-17:

☐ The course of study meets some criteria, but is lacking key components of a high quality course of study. We recommend the course be revised to include the following components prior to approval for Pilot Status in 2016-17.

☐ The course of study lacks all or most criteria of a high quality course of study. We recommend the course be rewritten with guidance and support from the district curriculum development staff.

Signatures
Reviewer 1: Reviewer 2: Reviewer 3:

If more space is needed for comments/feedback, attach separate page.

Sacramento
City Unified
School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Title of Course of Study: *Computer Science*

Title of Course for Instructional Materials Adoption: *N/A*

Please check one:

___School Used Only – **Identify school(s):**
X__District-Wide Use

Course of Study Reviewer & Title/Position: *Clayton Dagler, Math & Computer Science Teacher @Burbank*

Date Review Completed: *4/15/2016*

A. PROCESS/PROCEDURES

X__Curriculum Development
___Instructional Materials Adoption:

B. ALIGNMENT MATRIX

I. How does this course align to the California Content Standards and/or CA Common Core Standards?

It covers several standards in Common Core Math, Common Core English Language Arts Standards, and CTE / Information Communication Technology Standards

II. If the course is proposed as part of a Linked Learning Pathway or Career Academy, how does it align with the pathway and/or academy outcomes?

It can be the capstone class for an Information Technology pathway at Burbank.

III. How does the course engage students in academic work that meets the district's definition of academic rigor (academic rigor ="The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.")?

The course is aligned to a college level Computer Programming Class.

IV. Other Comments:

For Office Use Only---Approvals:

Dates each group approved the completed product:

- Principals:
- Assistant principals of instruction:
- High school curriculum committee:
- Cadre:
- Superintendent Cabinet:

Sacramento
City Unified
School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Reviewers' Feedback & Recommendation

Title of Course of Study:	*Computer Science*
Name and Position/Title of Reviewer 1:	*Roscoe Anh, Math and Computer Science Teacher @CK McClatchy HS*
Name and Position/Title of Reviewer 2:	*Clayton Dagler, Math and Computer Science Teacher @Luther Burbank HS*
Name and Position/Title of Reviewer 3:	*Joseph Stymeist, Interim Director of College and Career Readiness Department (former computer science teacher)*
Date Review Completed:	*4/15/2016*

Components and Criteria of a High Quality and Rigorous Course of Study:

- The course of study is a course within a clear and coherent sequenced of courses.
- The course of study is aligned to the CA Content Standards and/or CA Common Core Standards
- The course of study is aligned to the academy/pathway Program of Study and Student Learning Outcomes.
- The course engages students in academic work that meets the district's definition of Academic Rigor: *The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.*

☑ The course of study meets all criteria of a high quality course of study. We recommend it be approved for implementation beginning in 2016-17.

This course is based on the curriculum from the http://www.exploringcs.org/curriculum, which is most to date curriculum on computer science.

☐ The course of study meets most criteria of a high quality course of study. We recommend it be revised to include the following components prior to approval for implementation beginning in 2016-17:

☐ The course of study meets some criteria, but is lacking key components of a high quality course of study. We recommend the course be revised to include the following components prior to approval for Pilot Status in 2016-17.

☐ The course of study lacks all or most criteria of a high quality course of study. We recommend the course be rewritten with guidance and support from the district curriculum development staff.

Signatures

Reviewer 1: Reviewer 2: Reviewer 3:

If more space is needed for comments/feedback, attach separate page.

Sacramento City Unified School District

CURRICULUM DEVELOPMENT/ INSTRUCTIONAL MATERIALS ADOPTION

Title of Course of Study: ***Computer Science***

Title of Course for Instructional Materials Adoption: ***N/A***

Please check one:

___School Used Only – **Identify school(s):**
X__District-Wide Use

Course of Study Reviewer & Title/Position: ***Joseph Stymeist, Interim Director of College and Career Readiness Department***

Date Review Completed: ***4/14/2016***

A. PROCESS/PROCEDURES

X__Curriculum Development
___Instructional Materials Adoption:

B. ALIGNMENT MATRIX

I. How does this course align to the California Content Standards and/or CA Common Core Standards? Aligned

II. If the course is proposed as part of a Linked Learning Pathway or Career Academy, how does it align with the pathway and/or academy outcomes? It will prepare students for entry level work in the software development industry.

III. How does the course engage students in academic work that meets the district's definition of academic rigor (academic rigor ="The capacity to understand and apply content and experiences that are complex, ambiguous, provocative, and personally or emotionally challenging.")? Problem solving & critical thinking are required at a very high level in software development. Coding is one of the best methods for developing and honing these skills.

IV. Other Comments:
The fundamentals of computer programming are reinforced throughout rigorous exercises.

For Office Use Only---Approvals:

Dates each group approved the completed product:

- Principals:
- Assistant principals of instruction:
- High school curriculum committee:
- Cadre:
- Superintendent Cabinet: