SAN LEANDRO UNIFIED SCHOOL DISTRICT

COURSE PROPOSAL: Course of Study

SECTION A. COVER PAGE

1. **Course Title: Robotics I** **If Title Change, Previous Title:_____**	7. **Action** X New Course ▢ Course Revision ▢ Title Change Only
2. Date Submitted:	8. **Grade Level**(s) ▢ 6 ▢9 X 11 ▢ 7 X 10 X 12 ▢ 8
3. Transcript Title/Abbreviation(s)	**9. Prerequisites** (Please list.) ▢ Recommended Algebra I Geometry Product Design I
4. Transcript Course Code/Course Number :	**10. Seeking Program Distinction?** ▢ YES (Check one below.) X NO ▢Honors ▢ AP ▢Other
5. Subject Area CTE/ICT Pathway Concentrator	11. **Is this a CTE course?** X YES ▢ NO **If YES, complete Section C**
6. **Department** CTE	12. **Previously approved by UC as a course?** X YES ▢ NO ▢ NA for 6th-8th

13. UC ELIGIBILITY

Proposing eligibility?▢ NA for 6th-8th **XYES** ▢**NO**

a ▢ **b**▢ **c**▢ **d**▢ **e**▢ **f**▢ **g**X

14. **Unit Value/Course Credit/Length of Course**

▢ 0.5 (semester or half-year

X 1.0 (one year equivalent)

▢ 2.0 (two-year equivalent)

▢ Other

15. School Contact and Course Developer

Name: Andrew Doolittle

Phone: 510-618-4600 x2236

Email: adoolittle@slusd.us

Position: Computer Science Teacher/STREAM Academy Co-lead

16. Approval Signatures: **Approval Date:**

Department Chair 1/14/22

Principal: 1/14/22

DISTRICT OFFICE USE ONLY

Date approved by Board Curriculum Committee

1/24/22

Date approved by Board

SECTION B. COURSE CONTENT

17. Course Description

Robotics I introduces high school students to the working principles and foundations of computer science through robotics. Students learn to control robots using both block-based programming and text-based programming in “C” programming language. Virtual and hardware robots are used as platforms to engage students in personalized and collaborative learning of computer science integrated with science, technology, engineering, art, and math (STEAM) concepts. The course emphasizes hands-on robotics activities with a focus on human-computer interaction, computational thinking, and data analysis. It prepares students for solving math problems through coding. With creative design and extensive project-based activities, students develop critical thinking, problem solving, effective communication, and teamwork skills.

Robotics I will be the ICT Pathway concentrator course for the new STREAM academy, and will be taken by cohorts in their junior year.

18. Course Goals and/or Major Student Outcomes

Robotics I introduces high school students to the working principles and foundations of computer science through robotics. Students learn to control robots using both block-based programming and text-based programming in “C” programming language. Virtual and hardware robots are used as platforms to engage students in personalized and collaborative learning of computer science integrated with science, technology, engineering, art, and math (STEAM) concepts. As a result of taking this course, students will:

- Students understand the basic controls/features of the Linkbot and understand the RoboPlay Competition.
- Students can use a graphical user interface called Linkbot Labs to control the Linkbot.
- Students understand the basic robot programming using a C/C++ interpreter Ch.
- Students can use the RoboSim for robot simulation.
- Students can use basic programming features by using variables and generating Ch robot programs using RoboBlockly.
- Students can use input/output functions in robot programming, and utilize the number line for distance.
- Students can write programs to control a group of Linkbots to perform identical tasks such as dancing.
- Students can control a Linkbot configured as a two-wheel robot.
- Students can move a single robot in a Coordinate System.
- Students can write programs to control a single Linkbot with different motion characteristics.
- Students can write advanced programs to control a single Linkbot.
- Students can write programs to process the sensory information for Linkbots.

- Students can write programs to control multiple individual Linkbots.
- Students can move multiple robots in a Coordinate System.
- Students can write programs to control one or multiple group of Linkbots.
- Students can control multiple connected Linkbots.

19. Course Objectives (standards)

CTE Standards - ICT Sector - Software and Systems Development Pathway

- **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.
  - C1.3 Identify and describe how specifications and requirements are developed for new and existing software applications.
- **C4.0 Develop software using programming languages.**
  - C4.4 Identify and apply data types and encoding.
  - C4.6 Use proper programming language syntax.
  - C4.7 Use various data structures, arrays, objects, files, and databases.
  - C4.9 Create programs using control structures, procedures, functions, parameters, variables, error recovery, and recursion.
  - 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.**
  - C5.3 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.
- **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.
  - C9.5 Program a micro-controller for a device or robot.

20. Course Outline

Unit 1. Introduction to Coding and Robotics

This unit introduces students to block-based programming using the online RoboBlockly integrated learning environment. Students learn how to navigate and use the basic functionality of the RoboBlockly interface. Students also learn the basic terminology and principles of coding and robotics, addressing questions such as "What is coding?", "What is a computer language?", and "What is a robot?"

In order to explore the RoboBlockly environment, students will learn how to create a program that drives a virtual robot around a grid using driveDistance and turn commands.

Unit 2. Robot Programming

This unit expands the capabilities of students to program and control robots. Students will learn to use commands that control the virtual robot's trace line, speed, model (robot image), and size. They will learn new ways to move the robot by specifying the driving time, the (x,y) grid coordinates to drive to, the total angular rotation of the wheels, and/or the robot's orientation. Basic repeat loops will be used for repetitive robot movements, such as tracing a square. Wireless connection and control of hardware robots will then be introduced to give students the opportunity for hands-on learning. Finally, they will learn how to add and simultaneously control multiple robots in their programs, using both virtual and hardware robots and "non-blocking" commands.

Typical activities include having students program a robot to navigate an obstacle course or trace a figure, and program multiple robots to chase each other or perform a relay race. Both virtual robots and hardware robots will be used. Students will also gain practice and insights through debugging exercises.

Unit 3. Geometric Shapes, Colors, and Storytelling

In this unit students will learn to create and display geometric shapes, including lines, rectangles, quadrilaterals, triangles, circles, ellipses, arcs, stars, and polygons in different widths and colors. RGB colors with hexadecimal codes and the concepts of opacity and gradients will be covered. Students will also learn how to control the color and blinking of the LED on a robot, and how to change the background color or the background image of the RoboBlockly grid and use it for storytelling. Typical activities include having students use code to create geometric and artistic designs and create scenes and stories with virtual moving robots. The robots may be displayed as regular robots or as a variety of models (images), such as vehicles, animals, or people.

Unit 4. Language and Music

This unit introduces "say" and "think" commands that provide the capability to display cartoon bubbles with a robot, allowing students to create scenes with conversations among robots. The "speak" command will also be introduced, which enables the virtual robots to audibly speak the text students enter into the command in different voices and speeds. The "language" command will do automatic real-time translation of a robot's spoken words from one language to another,

such as from English text in the code to Spanish spoken words by the robot. Students will also learn about the basic types of musical notes and how to use RoboBlockly's musical capabilities, such as playNote and playMelody, as well as its virtual piano feature that can automatically generate playNote code as the piano is played by clicking on the keys. Robots can be programmed to move and play music at the same time.

Typical activities include creating musical compositions and more complex scenes and stories wherein the robots play music, "converse" with each other, and/or translate from one language to another.

Unit 5. Math, Data Analysis, and Plotting for Visualization

This unit starts by introducing students to variables and mathematical operations, including random numbers and statistical functions (mean, median, standard deviation). Students learn to use message boxes and print commands to display output, the "expr" command to plot mathematical expressions, and plotting options for colors, line widths, axis labels, legends, and the grid display. Students learn prompt commands to prompt for and store user input of numeric and string values. The concept of arrays is introduced, which enables students to organize and display data, do statistical calculations, and create 2D scatter plots and line plots.

Typical activities include writing programs to display, analyze, and plot data; prompt the user to enter values for the robot to act on, or generate random values to do so (e.g., enter the height and width of a rectangle the robot is to trace); and calculate the area of a triangle based on user input of the base and height.

Unit 6. Repetition with Counters, Decision Making, and Functions

This unit expands on the previously covered concept of a repeat loop by introducing count loops. The similarities and differences between repeat and count loops will be examined, and nested loops will be explored. Program branching will be introduced with conditional statements and their implementation in if, if-else, and if-elseif-else statements, along with relational and logical operators. The concepts of program modularization and code reuse will be introduced as students learn how to define and call their own functions.

Typical activities include using loops to program the robot to create "spirograph" figures, programming the robot to do one action or another based on the generation and testing of a random value, and creating functions to perform a calculation and return the result to the user.

Unit 7. Images and Animation

In this unit students will learn how to write code that displays and manipulates RoboBlockly's built-in images. Hundreds of images are available in over 40 categories, including people, buildings, furnishings, vehicles, animals, balls, balloons, clothes, and 2D and 3D images. Students will learn to size, flip, and rotate images, and upload new images. Using count loops (previously introduced), students will learn to create animations by continuously displaying, erasing, and redisplaying images in slightly different locations. Students will also learn how to create multiple successive scenes to tell a story involving animation or other robot motions, and to use

the RoboBlockly PreBoard for preprocessing of the code that creates the initial scene background.

Typical activities include creating animations using images and/or geometric shapes (such as a snowball increasing in size as it rolls down a hill).

Unit 8. Interactive Robotics, GUI, and Human-Computer Interaction

In this unit students will learn how to get the robot's (x,y) position, get the (x,y) location of a user's mouse click, and prompt the user to click in a rectangular region and test whether the user did so. Students also learn the principles of good graphical user interface (GUI) design and how to implement the GUI elements of radio buttons and checkboxes to prompt for user input and take appropriate actions for human-computer interaction.

Typical activities include writing programs to read the (x,y) location of a mouse click and then move the robot to that location, allow the user to control the actions of the robot by selecting from a menu of choices, or allow the user to create a scene by selecting images and potential locations from a menu of choices.

Unit 9. Introduction to C Programming

Students will be introduced to text-based programming using RoboBlockly's automatic translation of code blocks into C language code. Students will learn the basic syntax of C statements and explore arithmetic operations in the Ch Shell command line. They will create and run C programs generated by RoboBlockly.

Typical activities include prompting for user input, performing a calculation, and displaying the result (e.g., calculating the sales tax and total amount due based on the entered price of an item).

Unit 10. Creative Design and Projects

This final unit allows students to apply their creativity and newly developed coding and robotics skills to design and implement a program that utilizes many of the concepts and commands learned in the course. Students will also create projects in RoboBlockly with pre and post processing to express their creativity with innovative ideas. In addition, students have the option to participate in the RoboPlay Video Competition, a robotics-centric video competition sponsored by the C-STEM Center at the University of California, Davis. Students use their creativity in robotics, programming, math, writing, storytelling, art, music, choreography, design, video editing and film production in designing a well-organized visual performance. Students may also participate in the RoboPlay STEAM Competition using RoboBlockly.

21. Instructional Materials:

Core Instructional Materials:

Title: Learning Robot Programming with Linkbot for the Absolute Beginner
Edition: 8th
Publication Date: June 2019
Publisher: UC Davis C-STEM Center
Author(s): Harry H. Cheng
URL Resource(s): http://c-stem.ucdavis.edu

Supplementary Instructional materials:

Curriculum: Computer Science with Robotics in RoboBlockly Developer: Barobo, Inc.
Website: https://roboblockly.com/curriculum/

Title: Learning Computer Programming with Ch for the Absolute Beginner Edition: 2nd
Publication Date: May 2016
Publisher: UC Davis C-STEM Center
Author(s): Harry H. Cheng
URL Resource(s): http://c-stem.ucdavis.edu

Software: Ch Professional Edition
Developer: SoftIntegration, Inc.
Website: http://www.softintegration.com/download/

Software: C-STEM Studio
Developer: UC Davis C-STEM Center Website: https://c-stem.ucdavis.edu/studio/
Software: Linkbot Labs
Developer: Barobo, Inc.
Website: https://www.barobo.com/linkbotlabs

22. Instructional Methods and/or Strategies (Key Assignments)

Instructional Methods:

- Lecture
- Hands-on inquiry and exploration (Code Examination)
- Collaborative discussion
- Performance and competency based tasks (Remaking Software Platforms)

23. Assessment Methods and/or Tools

- Tests & Quizzes
- Formative Challenge Tasks
- End of Unit Performance Tasks

24. Grading Policy

Grading will follow the San Leandro Unified School District's policy.

SECTION C

25. Context for Course (optional)

REQUIRED FOR CTE COURSES

NOTE: In order to understand the context for a new course, sometimes it is helpful to understand the broader educational program and/or reform efforts of the school. How does this course fit into broader departmental and/or pathway structure? How does it fit into the overall school restructuring plans? Is the course intended to be a core course or supplemental? What are the student/school/community needs met by this course?

History of pathways at SLHS

STREAM Academy

- **STREAM Academy subject matter focus is on Robotics**
- **Academy is dual pathway - MPD (Manufacturing and Product Development > Product Innovation and Dev.) and ICT (Information Communication Technology > Software and Systems Development).**
- **Robotics I will be the concentrator class for the ICT/SSD pathway.**

Pathway course sequence is:

	MPD/PID	ICT/SSD
Sophomore	Product Design 1	
Junior		**Robotics 1**
Senior	Product Design 2	Robotics 2

26. History of Course Development (optional)

REQUIRED FOR CTE COURSES

NOTE: Likewise, it is sometimes helpful to know the origins of a course and who was involved in its development. Was this course modeled after another course at another school? If so, is that course UC approved? How does the course being submitted differ from the course after which it was modeled? Has this course received any special recognitions, designations or awards? Has it been articulated to local community colleges or universities?

This course is modeled after the UC Davis C-STEM Computer Science with Robotics course. We have adopted C-STEM as the basis for the STREAM Academy due to its coverage of relevant content, flexibility ot platform (live or remote, physical or virtual robots), suite of supporting applications and websites, and the advantage that C-STEM, as an UC-approved academic preparation program, gives students in priority applications to UC campuses.

For the initial course presentation, as a pilot in 2022-2023 school year, the course will be delivered as it is defined by UC Davis C-STEM. Based on that pilot modifications may be made for future school years.