Computer Science and Tech I

Course Overview for Computer Science and Technology I

The objective of this course is to introduce students to the breadth of the field of computer science through an exploration of engaging and accessible topics. The course is designed to focus on the conceptual ideas of computing and help students understand why certain tools or languages might be utilized to solve particular problems while also giving them a comprehensive understanding of the technologies used to build web applications. Primary areas of concentration include the computational practices of algorithm development, problem solving and programming within the context of problems relevant to today's students, interface design, limits of computers, and societal and ethical issues.

The Computational Thinking Practice Standards apply throughout the course and, together with the content standards, prescribe that students experience computer science as a logical, beautiful, and empowering subject that makes use of their ability to think critically and express themselves creatively.

CSTA K-12 CS Standards Emphasis: Understand the basics of computers and the internet · Explore and build basic models of intelligent behavior · Explore the societal impacts of computing · Interpret and analyze the efficiency algorithms and abstraction · Make connections between Mathematics and Computer Science · Explore and understand web-page design and development · Explore and understand programming design and development · Collect, generate, process, and visualize data and information · Integrate software and hardware through robotics

Course Competencies: for this course student will interface with the Computational Thinking Practices and College Board Computer Science Curriculum Requirements since the former allows students to understand the impacts of computing on society as well as draw connections between computing concepts, while the latter will set students up for more advanced courses in Computer Science.

Competency Sets:

- Computational Thinking Practices
    - Analyze the effects of the developments in computing
    - Design and implement creative solutions and artifacts
    - Apply abstractions and models
    - Analyze your own computational work and the work of others
    - Communicate computational thought processes, procedures, and results to others
    - Collaborate with peers on computing activities
- College Board Computer Science Curriculum Requirements

Course Modules: Throughout the course, students will navigate 6 modules adapted from the Exploring Computer Science Curriculum

Modules for this Course:

- *Module 1: Human Computer Interaction*
- *Module 2: Problem Solving*
- *Module 3: Web Design*
- *Module 4: Introduction to Programming*
- *Module 5: Computing and Data Analysis*
- *Module 6: Robotics*

Standards Coverage

Interim Session	Date Range	Standards Covered			
1	**Aug 29-Nov 3** (47 days)	CD.L2-03 CD.L3A-02 CD.L3B-02 CD.L3A-09 CD.L2-06 CPP.L2-01	CPP.L3A-07 CD.L2-01 CD.L2-02 CD.L2-04 CI.L2-01 CI.L2-02 CI.L2-03 CI.L2-05	CD.L2-01 CD.L2-07 CT.L2-03 CPP.L2-04 CT.L2-01 CT.L2-02 CT.L2-04	CT.L2-08 CT.L2-12 CT.L2-15 CT.L3A-01 CT.L3B-02 CT.L3B-03 CT.L3B-04
2	**Nov 7-Feb 3** (48 Days)	CPP.L3A-01 CPP.L3A-07	CPP.L2-03 CL.L2-02	CL.L3A-02 CPP.L3A-06	CPP.L3A-02 CPP.L3A-08
3	**Feb 6-Apr 12** (47 Days)	CPP.L3A-07 CT.L3B-06 CPP.L2-05 CT.L3B-10	CPP.L3A-12 CL.L2-03 CL.L3A-01 CL.L3B-01	CPP.L2-09 CPP.L3A-11 CPP.L3B-08 CT.L3A-04	CT.L3B-05 CPP.L3B-07 CT.L3B-06 CT.L3B-09
4	**Apr 24-June 30** (45 days)	CD.L2-02 CD.L2-03	CD.L2-06 CD.L2-07	CD.L2-08 CD.L3A-10	CD.L3B-05

Instructional Model and Implementation Strategies for Computer Science

Students will take Computer Science every other day for 90-minutes. The course will enable students to examine conceptual topics of computer science and computing while also acquiring skills necessary to dictate specific computing outcomes. The following strategies will be used to ensure students reach these outcomes:

Instructional Playlists: Students will be introduced to topics, skills, and tasks through playlists that include materials for a student to learn skills then practice them using various computing languages. Most playlists will include accountable video-based instruction, design nodes, and/or skills-based workshops that culminate in an artifact that exemplifies student completion. When appropriate students will be assessed on factual understanding of content as determined by the playlist curator.

Accountable video-based Instruction: videos used to introduce or review a specific computing concept or skill that includes practice problems and scenarios within the video ensuring that students attention and time spent. Student usage data is tracked as well as their achievement data on video-associated assessments.

Design Nodes: Students engage in a stepwise design process to achieve a large outcome. Students work through smaller outcomes (nodes) that when combined or sequenced together manifest the larger goal or outcome.

Skills-based Workshops: Students work through a self-paced skills workshop designed to lead students to acquire a specific skill. Workshops typically use a gamified or challenge-styled approach to building deepening levels of complexity around a skill.

Engineering and Design Thinking Challenges: students are provided with a challenge to program or code a solution, test prototypes, and design new models around a computing outcome. Students work independently and collaboratively to reach specific outcomes.

Methods of Assessment for Computer Science I

Performance Based Assessments: This course will utilize the Principled Assessment of Computational Thinking developed through a partnership of the National Science Foundation and SRI Education. It is aligned to the ECS Curriculum, CSTA Standards, and focuses on the Computational Thinking Practices. The assessment tools are to be used in the service of learning (not accountability) that provides teachers with the evidence they need to draw inferences of what students understand, know, and are able to do (also known as Knowledge, Skills, and Attributes or KSAs). They utilize scenario-based, multi-part short answer tasks; are linked to learning objectives of each module; contain multiple representations to support student understanding; and include multiple rubrics for each assessment along with additional materials to support the interpretation of student work.

Project Artifacts and Exhibitions: Throughout each interim students will be asked to investigate a challenge or problem aligned to the interim topic and complete a project that addresses the topic while also integrating skills learned in each course. Each project will have a course-specific artifact or exhibition that is assessed by the course teacher.

Core Knowledge and Comprehension Assessments: As the course provides students with factual or conceptual knowledge of a course's content they will be assessed on their mastery of knowledge and concepts. These assessments will emulate the college-board AP-styled exams for both AP Computer Science Principles and AP Computer Science A.

Module #1: Human Computer Interaction

In this module students are introduced to the concepts of computer and computing while investigating the major components of computers and the suitability of these components for particular applications. Students will experiment with internet search techniques, explore a variety of websites and web applications and discuss issues of privacy and security. Fundamental notions of Human Computer Interaction (HCI) and ergonomics are introduced. Students will learn that "intelligent" machine behavior is not "magic" but is based on algorithms applied to useful representations of information, including large data sets. Students will learn the characteristics that make certain tasks easy or difficult for computers, and how these differ from those that humans characteristically find easy or difficult. Students will gain an appreciation for the many ways in which computing-enabled innovations have had an impact on society, as well as for the many different fields in which they are used. Connections among social, economical and cultural contexts will be discussed.

Topic	# of Days	Standards	Description	Assessments
Exploring the concepts of computer and computing by investigating computer hardware components and a variety of internet resources	8	CD.L2-03 CD.L3A-02 CD.L3B-02 CD.L3A-09 CD.L2-06	Demonstrate an understanding of the relationship between hardware and software. Develop criteria for purchasing or upgrading computer system hardware. Identify and describe hardware (e.g., physical layers, logic gates, chips, components). Describe how the Internet facilitates global communication. Describe the major components and functions of computer systems and networks.	"Around the House" Students will be presented with several devices and must decide whether or not they fit the definition of a computer. Students must then write about any device they have at home that adheres to the definition of a computer. Students must be able to identify and explain the types of data it accepts, how it processes that data, and the result of that processing.
Exploring the use of computers in a variety of circumstances and fields	7	CPP.L2-01 CPP.L3A-07 CD.L2-01 CD.L2-02	Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems. Describe a variety of programming languages available to solve problems and develop systems. Recognize that computers are devices that execute programs. Identify a variety of electronic devices that contain computational processors	Performance-based Assessments (PBA)

		CD.L2-04 CI.L2-01 CI.L2-02 CI.L2-03 CI.L2-05	Identify a variety of electronic devices that contain computational processors Exhibit legal and ethical behaviors when using information and technology and discuss the consequences of misuse. Demonstrate knowledge of changes in information technologies over time and the effects those changes have on education, the workplace, and society. Analyze the positive and negative impacts of computing on human culture. Describe ethical issues that relate to computers and networks (e.g., security, privacy, ownership, and information sharing).	
The computer as a machine that needs to be provided specific instructions	7	CD.L2-01 CD.L2-07 CT.L2-03 CPP.L2-04	Recognize that computers are devices that execute programs. Describe what distinguishes humans from machines, focusing on human intelligence versus machine intelligence and ways we can communicate. Define an algorithm as a sequence of instructions that can be processed by a computer. Demonstrate an understanding of algorithms and their practical application.	"Activities a Robot Can Do" Students are provided with a scenario and asked to relate it to the types of tasks a robot can do. Students dissect the robot's instructions and decide then explain which instructions the robot would have difficulty completing. Students must be able to explain why an activity or task is or is not an example of a problem a computer can solve, describe or demonstrate the characteristics of a computer program, and describe the difference between intelligence as it relates to humans and computers.

Module #2: Problem Solving

This module provides students with opportunities to become "computational thinkers" by applying a variety of problem-solving techniques as they create solutions to problems that are situated in a variety of contexts. The range of contexts motivates the need for students to think abstractly and apply known algorithms where appropriate, but also create new algorithms. Analysis of various solutions and algorithms will highlight problems that are not easily solved by computer and for which there are no known solutions. This unit also focuses on the connections between mathematics and computer science. Students will be introduced to selected topics in discrete mathematics including Boolean logic, functions, graphs and the binary number system. Students are also introduced to searching and sorting algorithms and graphs.

	# of Days	Standards	Description	Assessments
Introduction and application of the Problem Solving Process	12	CT.L2-01	Use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, evaluation).	Performance-based Assessment (PBA)
		CT.L2-02	Describe the process of parallelization as it relates to problem solving.	
		CT.L2-04	Evaluate ways that different algorithms may be used to solve the same problem.	
		CT.L2-08	Use visual representations of problem states, structures, and data (e.g., graphs, charts, network diagrams, flowcharts).	
		CT.L2-12	Use abstraction to decompose a problem into sub problems.	
		CT.L2-15	Provide examples of interdisciplinary applications of computational thinking.	
		CT.L3A-01	Use predefined functions and parameters, classes and methods to divide a complex problem into simpler parts.	
		CT.L3B-02	Explain the value of heuristic algorithms to approximate solutions for intractable problems.	
		CT.L3B-03	Critically examine classical algorithms and implement an original algorithm.	
		CT.L3B-04	Evaluate algorithms by their efficiency, correctness, and clarity.	
		CL.L2-03	Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities.	

		CL.L3B-03 CPP.L2-04 CPP.L2-05	Evaluate programs written by others for readability and usability. Demonstrate an understanding of algorithms and their practical application. Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions.	
Binary Number System	4	CT.L2-14 CT.L3A-05	Examine connections between elements of mathematics and computer science including binary numbers, logic, sets and functions. Describe the relationship between binary and hexadecimal representations.	Performance-based Assessment (PBA)
Sorting Algorithms	10	CT.L2-04 CT.L2-05 CT.L2-08 CT.L3B-03 CT.L3B-04	Evaluate ways that different algorithms may be used to solve the same problem. Act out searching and sorting algorithms. Use visual representations of problem states, structures, and data (e.g., graphs, charts, network diagrams, flowcharts). Critically examine classical algorithms and implement an original algorithm. Evaluate algorithms by their efficiency, correctness, and clarity.	"Clubs Item" Students are provided with two algorithms. They must enact the algorithms and then compare them. Students are given a scenario where a principal has to find out the best method of placing students into clubs of their choice based on their preference lists. Students will compare 3 different methods a principal can go about doing this then choose the most effective algorithm while providing explanations.

				Students must be able to state what an algorithm would output given a set of inputs, evaluate the extent to which an algorithm solves the stated problem, and compare the tradeoffs between different algorithms for solving the same problem.

Module #3: Web Design

This section prepares students to take the role of a developer by expanding their knowledge of algorithms, abstraction, and web page design and applying it to the creation of web pages and documentation for users and equipment. Students will explore issues of social responsibility in web use. They will learn to plan and code their web pages using a variety of techniques and check their sites for usability. Students learn to create user-friendly websites. Students will apply fundamental notions of Human Computer Interaction (HCI) and ergonomics.

Topic	# of Days	Standards	Description	Assessments
Introduction to Websites and HTML	6	CPP.L3A-01 CPP.L3A-07 CPP.L2-03 CL.L2-02 CL.L3A-02	Create and organize web pages through the use of a variety of web programming design tools. Describe a variety of programming languages available to solve problems and develop systems. Design, develop, publish, and present products (e.g., webpages, mobile applications, animations) using technology resources that demonstrate and communicate curriculum concepts. Collaboratively design, develop, publish, and present products (e.g., videos, podcasts, websites) using technology resources that demonstrate and communicate curriculum Concepts. Use collaborative tools to communicate with project team members (e.g., discussion threads, wikis, blogs, version control, etc.).	Performance-based Assessment (PBA)
Image editing (with Photoshop), and Styling with CSS	7	CPP.L3A-06	Select appropriate file formats for various types and uses of data.	Performance-based Assessment (PBA)
Intermediate CSS - Page Layouts, User Interface Elements, etc.	17	CPP.L3A-02	Use mobile devices/ emulators to design, develop, and implement mobile computing applications.	"Designing a webpage" Students are provided with a general idea for a webpage and must discuss design elements

				and layout of the webpage. Students must be able to create a set of specifications for a web page given the intent of the web page and then design a webpage based on specified objectives (web pages will be focused on a practical, personal, and/or societal purpose)..

Module #4: Introduction to Programming

Students are introduced to some basic issues associated with program design and development. Students design algorithms and create programming solutions to a variety of computational problems using an iterative development process in Scratch. Programming problems include mathematical and logical concepts and a variety of programming constructs.

Topic	# of Days	Standards	Description	Assessments
Introduction to Programming using Blockly Programming	8	CPP.L3A-07 CPP.L3A-08	Describe a variety of programming languages available to solve problems and develop systems. Explain the program execution process.	Performance-based Assessment (PBA)
Introduction to Variables, Conditionals, and Loops	26	CPP.L3A-07 CT.L3B-06 CPP.L2-05 CT.L3B-10 CPP.L3A-12 CL.L2-03 CL.L3A-01 CL.L3B-01	Describe a variety of programming languages available to solve problems and develop systems. Compare and contrast simple data structures and their uses (e.g., arrays and lists). Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions. Decompose a problem by defining new functions and classes. Describe how mathematical and statistical functions, sets, and logic are used in computation. Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities. Work in a team to design and develop a software artifact. Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.	"Gabriela's and Lucia's Algorithm" Students are provided with two algorithms. They must enact the algorithms and then compare them. They also must describe how programming structures could be used when programming the algorithms. Students must be able to state what an an algorithm would output given a set of inputs, compare the tradeoffs between different algorithms for solving the same problem, evaluate the extent/degree to which a program solves the stated problem, describe features of a programming

				structure, and evaluate the relationship between features of a programming structure and features of a problem or algorithm.

Module #5: Computing and Data Analysis

In this unit students explore how computing has facilitated new methods of managing and interpreting data. Students will use computers to translate, process and visualize data in order to find patterns and test hypotheses. Students will work with a variety of large data sets that illustrate how widespread access to data and information facilitates identification of problems. Students will collect and generate their own data related to local community issues and discuss appropriate methods for data collection and aggregation of data necessary to support making a case or facilitating a discovery.

Topic	# of Days	Standards	Description	Assessments
Data Collection and validating claims with data	10	CPP.L2-09 CPP.L3A-11 CPP.L3B-08	Collect and analyze data that is output from multiple runs of a computer program. Describe techniques for locating and collecting small and large-scale data sets. Deploy various data collection techniques for different types of problems.	Performance-based Assessment (PBA)
Data Analysis Techniques	25	CT.L3A-04 CT.L3B-05 & CPP.L3B-07 CT.L3B-06 CT.L3B-09	Compare techniques for analyzing massive data collections. Use data analysis to enhance understanding of complex natural and human systems Compare and contrast simple data structures and their uses (e.g., arrays and lists). Analyze data and identify patterns through modeling and simulation.	Performance-based Assessment (PBA)

Module #6: Robotics

This unit introduces robotics as an advanced application of computer science that can be used to solve problems in a variety of settings from business to healthcare and how robotics enables innovation by automating processes that may be dangerous or otherwise problematic for humans. Students explore how to integrate hardware and software in order to solve problems. Students will see the effect of software and hardware design on the resulting product. Students will apply previously learned topics to the study of robotics.

Topic	# of Days	Standards	Description	Assessments
The features of robots	4	CD.L2-02 CD.L2-03 CD.L2-06 CD.L2-07 CD.L2-08	Identify a variety of electronic devices that contain computational processors. Demonstrate an understanding of the relationship between hardware and software. Describe the major components and functions of computer systems and networks. Describe what distinguishes humans from machines, focusing on human intelligence versus machine intelligence and ways we can communicate. Describe ways in which computers use models of intelligent behavior (e.g., robot motion, speech and language understanding, and computer vision).	Performance-based Assessment (PBA)
Familiarization with the robot(s) and their software	13	CD.L2-02 CD.L2-03 CD.L2-06	Identify a variety of electronic devices that contain computational processors. Demonstrate an understanding of the relationship between hardware and software. Describe the major components and functions of computer systems and networks.	Performance-based Assessment (PBA)
Robotics Projects	23	CD.L3A-10 CD.L3B-05	Describe the major applications of artificial intelligence and robotics. Explain the notion of intelligent behavior through computer modeling and robotics.	Performance-based Assessment (PBA)