

Course Title:	Introduction to Programming
Grade(s):	9-12

Course Description
Introduction to Programming is a foundational computer science course that introduces students to the principles of coding, computational thinking, and problem solving. Through hands-on programming activities and project-based learning experiences, students learn how software is designed, developed, tested, and improved. Students will use a modern programming language to create programs, analyze data, solve real-world problems, and develop digital solutions. Emphasis is placed on logical reasoning, creativity, collaboration, and ethical technology use. No prior programming experience is required. Emerging technologies, including Artificial Intelligence, are integrated throughout the units of study. This course aligns with the Computer Science Teachers Association (CSTA) K-12 Computer Science Standards and supports the development of skills identified by employers and postsecondary institutions as essential for success in STEM and technology-related fields.
Approved Curricular Materials/Textbook(s)
Code.org Google Colab

Unit Title	Unit Description (Include Unit Objectives)	Unit Pacing
Unit 1: Introduction to Computer Science and Programming	Students begin the course by exploring the fundamental concepts of computer science and the role technology plays in modern society. They learn how computers process information, examine the relationship between hardware and software, and investigate career pathways within computing and technology fields. Students are introduced to computational thinking strategies such as decomposition, pattern recognition, abstraction, and algorithm design. They begin writing simple programs using variables, data types, operators, and user	3 weeks

	input/output to solve basic problems and communicate information. Standards Alignment: CSTA 3A-CS-01, CSTA 3A-IC-25, CSTA 3A-AP-16	
Unit 2: Algorithms, Decision Making, and Problem Solving	Students learn how programmers design solutions using algorithms and logical reasoning. Through flowcharts, pseudocode, and coding activities, students develop programs that utilize Boolean logic, comparison operators, and conditional statements. They create interactive programs capable of making decisions based on user input and environmental conditions. Emphasis is placed on program design, debugging, testing, and refining solutions to improve functionality and user experience. Standards Alignment: CSTA 3A-AP-17, CSTA 3A-AP-21, CSTA 3A-AP-22	4 weeks
Unit 3: Iteration, Functions, and Modular Programming	Students expand their programming skills by exploring iteration and modular design. They learn how loops can automate repetitive tasks and make programs more efficient. Students use functions and procedures to organize code into reusable components, improving readability and maintainability. Through increasingly complex coding challenges, students develop programs that combine decision-making, iteration, and modular structures to solve authentic problems. Debugging, documentation, and coding best practices are emphasized throughout the unit. Standards Alignment: CSTA 3A-AP-18, CSTA 3A-AP-21, CSTA 3A-AP-22	5 weeks
Unit 4: Data and Computing	In this unit, students explore how computers collect, store, organize, and analyze data. They work with lists and other data structures to process information and solve data-driven problems. Students investigate how data is used in fields such as healthcare, business, science, and education. Through project-based learning,	4 weeks

	students create simple data visualizations, analyze trends, and communicate findings. Discussions also address data privacy, cybersecurity, and the ethical use of information in a digital society. Standards Alignment: CSTA 3A-DA-09, CSTA 3A-DA-11, CSTA 3A-IC-25	
Unit 5: Software Development Capstone Project	The semester culminates in a comprehensive capstone project that allows students to apply the programming concepts and computational thinking skills developed throughout the course. Students identify a problem or need, design a software-based solution, and follow a structured development process that includes planning, coding, testing, debugging, revision, and documentation. Students present their completed projects to peers and stakeholders while reflecting on their growth as programmers and problem solvers. Standards Alignment: CSTA 3A-AP-23, CSTA 3A-AP-24, CSTA 3A-AP-22	5-6 weeks