Collaborate to create mobile apps. Solve problems and create value for others through innovation and creativity. Explore how innovations in computing impact and connect our world. With a gentle introduction to programming, you will learn how to put your designs into practice.

Whether these are your first steps in computer science, or a continuation of your journey, Computer Science Essentials will give you confidence to succeed today and beyond.

Computer Science Essentials is designed to be a full-year (180-day) course implemented in the 9th grade. This course is an excellent entry point for new high school computer science (CS) learners. And students who have prior CS experiences will find ample opportunity to expand upon those experiences in this course. All students who take CS Essentials will have many opportunities for creative expression and exploration in topics of personal interest, whether it be through app development, web design, or connecting computing with the physical world. CS Essentials is designed with strong connections to the Computer Science K12 Frameworks (CS K12), the Computer Science Teachers Association K-12 Computer Science (CSTA K-12 CS) Level 3A Standards, and the Advanced Placement Computer Science Principles (AP CSP) Frameworks. Though CS Essentials is not an instance of the AP Computer Science Principles course (CSP), it will boost student success for those who continue in CS courses. These intentional connections to widely accepted standards will help students gain confidence and reinforce essential concepts and skills that build toward life-long success in the computer science pathways beyond just PLTW courses.

CS Essentials introduces students to coding fundamentals through an approachable, block-based programming language where they will have early success in creating usable apps. As students sharpen their computational thinking skills, they will transition to programming environments that reinforce coding fundamentals by displaying block programming and text based programming side-by-side. Finally, students will learn the power of text-based programming as they are introduced to the Python® programming language.

The course engages students in computational thinking practices and collaboration strategies, as well as industry-standard tools authentic to how computer science professionals work. Students will learn about professional opportunities in computer science and how computing can be an integral part of all careers today. The following is a list of the units of study in the course.

    Unit 1   Creative Computing: Building with Blocks
    Unit 2   Computing and Society: Transitions to Text
    Unit 3   Web Development: Solving with Syntax
    Unit 4   Computing with a Purpose

Unit 1: Creative Computing: Building with Blocks

Unit 1 welcomes new and returning students to the world of computer science and coding fundamentals. Students work with MIT App Inventor to create basic apps that rely on the concepts of event-driven programming, branching, iteration, variables, and abstraction—the building blocks of creating with code. Students are introduced to essential computational thinking practices, such as developing abstractions, collaborating around computing,

and communicating as they create, test, and refine computational artifacts of Android™ apps.

Creative Computing: Building with Blocks
Lesson 1.1   Introduction to Computer Science Essentials and Coding Fundamentals
Lesson 1.2   Collaborating Around Computing
Lesson 1.3   Innovation and Computational Problem Solving

Lesson 1.1 Introduction to Computer Science Essentials and Coding Fundamentals
Mobile computing has changed our world, and many of today's students have never known a life without apps. This lesson gives students the tools they need to create their own apps using MIT App Inventor. The goal of this lesson is to introduce students to coding fundamentals through block-based programming. Students will develop independent and collaborative strategies that will help them communicate around computing as they learn and reinforce the fundamental concepts of coding. With a powerful yet approachable tool, students will use their creativity to produce computational artifacts like those that are essential to all of us today.

Activity 1.1.1   Block-based Programming: Getting Started with MIT App Inventor
Activity 1.1.2   Conditionals and Event-driven Programming
Activity 1.1.3   Operators, Data Types, Logic, and Strings
Activity 1.1.4   Global Variables and Local Variables
Activity 1.1.5   Pair Programming: Iteration and Loops
Project 1.1.6   App Development: Create a Game

Lesson 1.2 Collaborating Around Computing
This lesson focuses on collaborative strategies that coding professionals use when creating programs and applications, while it continues to introduce essential concepts in computer science and coding. The lesson also introduces the idea that computer science can be more than just innovation and creative expression; it can be powerful in trying to solve many problems in today's world. Students apply an Agile development process and task decomposition to solve a problem that meets the needs of others.

Activity 1.2.1   Problem Solving and Innovation
Activity 1.2.2   Algorithms
Activity 1.2.3   Procedural Abstraction: Procedures, Functions, and Methods
Activity 1.2.4   Lists
Project 1.2.5   Decomposition: Solve a Problem

Lesson 1.3 Innovation and Computational Problem Solving
The final lesson of this unit gives students the freedom to select the focus of their development in choosing the type of app they would like to collaborate to create. Student groups will apply development strategies and user-centered research to create an app that has value to others. Students will gain insight on the importance of creativity, persistence, and value of diverse perspectives in an iterative development process.

Problem 1.3.1   App Development: Creating Value for Others

Unit 2: Computing and Society: Transitions to Text

Unit 2 continues to reinforce coding fundamentals as students are gradually introduced to text-based programming. In this unit, students will explore the impacts of computer science on our society and bring coding off the screen and into the physical world. Students will learn how images can be used to make decisions in programs and explore real-world applications and innovations that will shape our future.

Computing and Society: Transitions to Text
Lesson 2.1   Transitions to Text-based Coding
Lesson 2.2   Computing and Careers
Lesson 2.3   Computing in Our World

Lesson 2.1 Transitions to Text Based Coding
Block-based programming is a great way to introduce coding fundamentals, but many students want to know, "What is happening inside those blocks?" Lesson 2.1 introduces students to the idea of a lower level of abstraction in a programming language. Students will develop in an environment that allows them to create in blocks, but see that same code in a text-based language.

Activity 2.1.1   Transitioning from Blocks to Text
Activity 2.1.2   Variables, Strings, Operators, and Conditions
Activity 2.1.3   Functions and Loops
Activity 2.1.4   Lists
Activity 2.1.5   Procedure-oriented Programming
Activity 2.1.6   Object-Oriented Programming
Project 2.1.7   Automating Our World

Lesson 2.2 Computing and Careers
Just as clicks of a button or "swipes" of a screen are used to trigger events in an app, today, images are becoming increasingly important as a way to make decisions in programming. In this lesson, students will explore image processing and other innovations that are changing our society. Students will also begin to investigate the wide range of careers in computer science and how computational thinking is an important part of the majority of professions today and in the future.

Activity 2.2.1   Careers in Computer Science
Activity 2.2.2   Big Data and Image Processing
Activity 2.2.3   Image Processing Methods
Activity 2.2.4   Decisions from Images
Project 2.2.5   Image Processing: Navigation and Collision Avoidance

Lesson 2.3 Computing in Our World
Tomorrow's solutions involve all of us. In the final lesson, student groups will learn how to take collaborations to scale to achieve a common goal.

Project 2.3.1   Cooperative Driving and Self-driving Vehicles

Unit 3 Web Development: Solving with Syntax

The goal of Unit 3 is for students to begin to understand and use the flexibility and power of programming in a text-based environment. They will learn how client-side and server-side connections make the Web work. Students will be introduced to the Python® programming language in the collaborative Cloud9 development environment. In this unit, students will continue to build on coding fundamentals as they apply the same coding concepts, computational thinking practices, and development processes introduced in units 1 and 2.

Web Development: Solving with Syntax
Lesson 3.1   The Power of Text-based Coding
Lesson 3.2   Web Languages and the Internet
Lesson 3.3   Creating and Connecting

Lesson 3.1 The Power of Text-Based Coding

In this lesson, students will reinforce previously learned concepts as they are introduced to the power of programming in a text-based language. The goal of this lesson is for students to become comfortable implementing algorithms using conditionals and loops in Python. Students create a game simulation and reinforce what they have learned about functions, arguments, and return values. Students generalize from this simulation to learn about model abstraction and the impact that simulation and data are having across career fields. Students then apply their Python skills to compete in a rock-paper-scissors game, developing functions to implement a complex strategy that attempts to detect and react to their opponent's strategy.

Activity 3.1.1   Python Programing on Cloud9
Activity 3.1.2   Variables and Conditionals
Activity 3.1.3   Data Types, Lists, and Elements
Activity 3.1.4   Iteration and Counts
Activity 3.1.5   Built-in Functions
Project 3.1.6   PS Rock

Lesson 3.2 Web Languages and the Internet

In this lesson students will be introduced to the Django web development platform as they explore how websites are designed and viewed on the client side, as well as how they connect, manage, and share data on the server side. Students will be exposed to common web programming languages as they learn to make decisions about how people view and interact with a website and how to protect data that is exchanged.

Activity 3.2.1   The Web, Crowds, and Clouds: Explore Task
Activity 3.2.2   HTML and Basic Web Page Formatting
Activity 3.2.3   Configuring a Server
Activity 3.2.4   Setting Up Databases
Activity 3.2.1   Creating Web Forms and Views
Project 3.2.5   Web Development: Creating and Connecting a Website

Lesson 3.3 Creating and Connecting Websites

This lesson will allow students to collaboratively design, create, and connect a secure website based on an

interest or need that the student group defines. Student groups will apply development strategies and user-centered research to create a website that has value to others and protects sensitive data.

Problem 3.3.1   Web Development: Creating Your Own Website

Unit 4 Computing with a Purpose

The final unit in CS Essentials allows students to apply all that they have learned in a student-defined, student-driven development. Whether creating an app, a website, or a physical computing device, students will apply computational thinking practices and a strategic development process to create computational artifacts that solve problems and create value for others. Students will collaborate the way computing professionals do as they pursue solutions to authentic needs. For those students continuing on to CSP, this unit provides an excellent model of how to participate in, document, and create a performance task for AP CSP.

Computing with a Purpose
Lesson 4.1   Innovation of Computational Problem Solving

Lesson 4.1 Innovation of Computational Problem Solving
The goal of this lesson is to allow students the opportunity to apply the collaboration, technical, and communication skills that they have developed to solve an authentic problem that is relevant to them.

Problem 4.1.1   Your Development Process: Create Performance Task
    Part a   Finding a Development to Pursue
    Part b   Presenting Your Development Process
    Part c   Prepare, Investigate, and Plan
    Part d   Design, Create, and Test
    Part e   Evaluation and Reflection