TOWNSHIP OF UNION PUBLIC SCHOOLS

Computer Science Essentials

Approved November 20, 2018

Updated December 18, 2018

Mission Statement

The mission of the Township of Union Public Schools is to build on the foundations of honesty, excellence, integrity, strong family, and community partnerships. We promote a supportive learning environment where every student is challenged, inspired, empowered, and respected as diverse learners. Through cultivation of students' intellectual curiosity, skills and knowledge, our students can achieve academically and socially, and contribute as responsible and productive citizens of our global community.

Philosophy Statement

The Township of Union Public School District, as a societal agency, reflects democratic ideals and concepts through its educational practices. It is the belief of the Board of Education that a primary function of the Township of Union Public School System is to formulate a learning climate conducive to the needs of all students in general, providing therein for individual differences. The school operates as a partner with the home and community.

Recommended Program:

Project Lead the Way – Computer Science Essentials

Curriculum Units

Unit 1 Creative Computing: Building with Blocks (25%)
Unit 2 Computing and Society: Transitions to Text (25%)
Unit 3 Solving with Syntax (35%)
Unit 4 Computing with a Purpose (15%)

Course Description

With emphasis on computational thinking and collaboration, this year-long course provides an excellent entry point for students to begin or continue the PLTW Computer Science K-12 experience. Computer Science Essentials will expose students to a diverse set of computational thinking concepts, fundamentals, and tools, allowing them to gain understanding and build confidence.

In Computer Science Essentials, students will use visual, block-based programming and seamlessly transition to text-based programming with languages such as Python to create apps and develop websites, and learn how to make computers work together to put their design into practice. They'll apply computational thinking practices, build their vocabulary, and collaborate just as computing professionals do to create products that address topics and problems important to them.

Computer Science Essentials helps students create a strong foundation to advance to Computer Science Principles, Computer Science A, and beyond.

I. CSE Units (Overview, Pacing Guide, Standards, Description, Activities)

Unit 1 Creative Computing: Building with Blocks (45 days)

Summary: 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 Unit Summary

Lesson 1.1	Introduction to Computer Science Essentials (19 days)
Lesson 1.2	Collaborating Around Computing (17 days)
Lesson 1.3	Innovation and Problem Solving (9 days)

Lesson 1.1 Introduction to Computer Science Essentials

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	Getting Started with Block-Based Programming: Digital Doodle (3 days)
Activity 1.1.2	Algorithms and Coding Fundamentals: Happy Accelerometer (3 days)
Activity 1.1.3	Conditionals and Event-Driven Programming: Happy Balance (2 days)
Activity 1.1.4	Local and Global Variables: Guessing Game 2 Player (3 days)
Activity 1.1.5	Iteration and Loops: Guessing Game 1 Player (3 days)

Project 1.1.6 App Development: Creative Expression (5 days)

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: Interview Database (2 days)

Activity 1.2.2 Algorithms and APIs: Hack Attack (3 days)

Activity 1.2.3 Procedural Abstraction: Price per Slice (3 days)

Activity 1.2.4 Lists: Survey Says (3 days)

Project 1.2.5 App Development: Problem Solving and Innovation (6 days)

Lesson 1.3 Innovation and 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 (9 days)

Unit I Standards: Competencies (C), Domains (D), Objectives (O), Knowledge and Skills (KS)

C1 Creativity and Problem Solving

Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.

D1 Creativity

Computing is a creative activity. Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact.

O1.1 Apply a creative development process when creating computational artifacts. LO 1.1.1 [P2]

KS1.1.1 Translate ideas into tangible form by creating computational artifacts and employing an iterative and exploratory process. EK 1.1.1B

D2 Problem-Solving Mindset

There are professional characteristics and habits of action that help people create value for society through innovation and problem solving.

O2.1 Describe moments within a process where curiosity, persistence, and the positive aspect of failure played an important role in gaining understanding about a problem or unexpected observation.

KS2.1.1 (2b IWR) Describe difficulties and/or opportunities you encountered and how they were resolved or incorporated.

O2.2 Engage stakeholders in a problem and use their perspectives to shape the course of your development.

KS2.2.1 Identifying programmer and user concerns that affect the solution to problems. EK 5.1.2G

D3 Problem-Solving Process

A computational problem solving process is an iterative, systematic approach by which a team generates and validates a proposed solution.

O3.1 Apply and describe an iterative process based on user-centered research to solve a problem.

KS3.1.1 Apply and describe an iterative process used during the development of a solution.

O3.2 Identify and apply decomposition as a critical step in problem solving.

KS3.2.1 Deconstruct a complex project or problem into smaller discrete modules that can be developed independently, then incorporated together at a later time.

KS3.2.2 Deconstruct a complex problem into simpler parts using predefined constructs (e.g., functions and parameters and/or classes). 3A-A-4-8

O3.3 Explain how people participate in a problem-solving process that scales. LO 7.1.2

KS3.3.1 Describe how human capabilities are enhanced by digitally enabled collaboration.
KS3.3.2 Explain how services use the contributions of many people to benefit both individuals and society.

D4 Computational Tools and Techniques

Computing involves the application of collaboration tools, programming tools, mathematical principles, and techniques to manage developments.

O4.1 Select and apply appropriate computational tools and techniques to solve a problem or create value for others.

KS4.1.1 Select tools for collaborating for data collection, writing, or programming.
KS4.1.2 Gain understanding of software tools and services while creating computational artifacts. EK 1.2.1B

O4.2 Apply a system of version control effectively.

KS4.2.1 Maintain successive versions of a digital product during development.

Unit 2: Computing and Society: Transitions to Text (45 days)

Summary: Unit2 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 Unit Summary

Lesson 2.1 Transitions to Text-based Coding (16 days)

Lesson 2.2 Computing and Careers in our Society (19 days)

Lesson 2.3 Computing in Our World (10 days)

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 (3 days)

Activity 2.1.2 Coding Fundamentals: Dead Reckoning (3 days)

Activity 2.1.3 Coding Fundamentals: Arrays (3 days)

Project 2.1.4 Wavefront Navigation (7 days)

Lesson 2.2 Computing Innovations 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, Innovation, and Ethics in Computer Science (4 days)

Activity 2.2.2 Image Processing: Identification (4 days)

Activity 2.2.3 Decisions from Images (4 days)

Project 2.2.4 Image Processing: Navigation and Collision Avoidance (7 days)

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 Create Performance Task: Cooperative Driving (10 days)

Unit 2 Standards: Competencies (C), Domains (D), Objectives (O), Knowledge and Skills (KS)

C2 Technical Knowledge and Skills

Every career field requires technical literacy and career-specific knowledge and skills to support professional practice.

D5 Data

Data and information facilitate the creation of knowledge. Managing and interpreting an overwhelming amount of raw data is part of the foundation of our information society and economy.

O5.1 Collect, organize, and explore real and simulated data.

KS5.1.1 Identify tools and creative methods to collect and process data.

D6 Algorithms

Algorithms are used to develop and express solutions to computational problems. Algorithms are fundamental to even the most basic everyday task.

O6.1 Develop an algorithm for implementation in a program. LO 4.1.1 [P2]

KS6.1.1 Understand that sequencing, selection, and iteration are building blocks of algorithms. EK 4.1.1A

KS6.1.2 Understand that sequencing is the application of each step of an algorithm in the order in which the statements are given. EK 4.1.1B

O6.2 Express an algorithm in a language. LO 4.1.2 [P5]

KS6.2.1 Contrast the languages for algorithms including natural language, pseudocode, and visual and textual programming languages. EK 4.1.2A

D7 Abstraction

Abstraction reduces information and detail to facilitate focus on relevant concepts. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems.

O7.1 Describe the variety of abstractions used to represent data. LO 2.1.1 [P3]

KS7.1.1 Understand that high-level programming languages provide more abstractions for the programmer and make it easier for people to read and write a program. EK 2.2.3B

Unit 3 Solving with Syntax (55 days)

Summary: 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. 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.

Solving with Syntax Unit Summary

Lesson 3.1 Collaborating in Text

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.

Lesson 3.2 Text-based Solutions

In this lesson, students will continue to explore the use of text-based programming. The lesson ends with students creating a game simulation that allows them to make generalizations and develop functions that attempt to detect and react to another team's strategy.

Activity 3.2.1 Data Types, Lists, and Elements (4 days)

Activity 3.2.2 Social Media Posts (6 days)

Activity 3.2.3 Iteration and Counts (5 days)

Activity 3.2.4 Course Registration (7 days)

Project 3.2.5 Artificial Intelligence: Rock, Paper, Scissors Simulation (8 days)

Lesson 3.3 The Power of Text-based Programming

In the final unit of this lesson, students will work with a team to create a program that automates the solution of a problem from one of their other classes.

Problem 3.3.1 From Paper to Programming (16 days)

Unit 3 Standards: Competencies (C), Domains (D), Objectives (O), Knowledge and Skills (KS)

C2 Technical Knowledge and Skills

Every career field requires technical literacy and career-specific knowledge and skills to support professional practice.

D8 Programming

Programming enables problem solving, human expression, and creation of knowledge. Any particular programming language is selected based on appropriateness for a specific project or problem.

O8.1 (2a IWR) Creative Expression in Programming - Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. LO 5.1.1 [P2] Or (2b IWR)Problem Solving in Programming - Develop a program to solve problems. LO 5.1.

KS8.1.1 Develop programs used in a variety of ways by a wide range of people. EK 5.1.1A

KS8.1.2 Understand that programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may have visual, audible, or tactile inputs and outputs. EK 5.1.1B

O8.2 Iteration in Programming - Create programs by writing and testing code in a modular, incremental approach.

KS8.2.1 (2b IWR) Describe how an iterative process of program development helps in developing a correct program to solve problems. EK 5.1.2A

KS8.2.2 Incrementally add tested program segments to correct working programs to help create larger correct programs. EK 5.1.2C

KS8.2.3 Adapt or improve existing code.

O8.3 Algorithms in Programs - Explain how programs implement algorithms.

O8.4 Abstraction in Programs - Use an abstraction to manage complexity in programs. LO 5.3.1 [P3]

KS8.4.1 Understand that procedures are reusable programming abstractions. EK 5.3.1A

KS8.4.2 Understand that a procedure is a named grouping of programming instructions. EK 5.3.1B

KS8.4.3 Use procedures to reduce the complexity of writing and maintaining programs. EK 5.3.1C

KS8.4.4 Understand that procedures have names and may have parameters and return values.

Unit 4 Computing with a Purpose (35 days)

Summary: The final unit in CSE 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 PLTW 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 Unit Summary

Lesson 4.1 Innovation of Computational Problem Solving (35 days)

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 Create Performance Task: Your Development Process

- Part A Find an Idea to Pursue (4 days)
- Part B Set Your Development Milestones (1 day)
- Part C Prepare, Investigate, and Plan (3 days)
- Part D Design, Create, and Test (15 days)
- Part E Evaluate and Reflect (7 days)
- Part F Present (5 days)

Unit 4 Standards: Competencies (C), Domains (D), Objectives (O), Knowledge and Skills (KS)

C3 Professional Practices and Communication

Professional practice is guided by professional ethics and standards and requires effective communication and collaboration.

D13 Career Awareness

Today computing impacts almost all careers. There are career specializations within computer science such as software development, security, network, and systems administration.

O13.1 Describe career paths within the computing specialties.

II. Technology and ELD Standards

Technology Standards; English Language Development Standards		
Technology Standards in Each Unit 8.1.8.A.1 Demonstrate knowledge of a real world problem using digital tools. 8.1.12.C.1 Develop an innovative solution to a real world problem or issue in collaboration with peers and experts, and present ideas for feedback through social media or in an online community 8.1.2.B.1 Illustrate and communicate original ideas 8.1.12.D4 Research and understand the positive and negative impact of one's digital footprint. 8.1.8.A.4 Graph and calculate data within a spreadsheet and present a summary of the results 8.1.12.D.5 Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.	**ELD** **Standard 2: Language of Language Arts:** English Language Learners communicate information, ideas and concepts necessary for academic success in the content area of language arts. **Standard 3: Language of Mathematics:** English Language Learners communicate information, ideas and concepts necessary for academic success in the content area of mathematics. **Standard 4: Language of Science:** English Language Learners communicate information, ideas and concepts necessary for academic success in the content area of science.	

III. Differentiation/Modifications for Teaching

Research-Based Effective Teaching Strategies	Modifications	Special Education	Strategies for English Language Learners
Questioning techniques to facilitate learning ~ See also **Five Practices for Orchestrating Math Discussion** Math Discourse ~ 5 talk Moves Constructivist learning opportunities ~Piaget, Vygotsky, Bruner Promote linguistic and nonlinguistic representations Cooperative Learning Various types of feedback ~Student to student feedback; Teacher to student feedback Varied opportunities for students to communicate (orally, writing) Use technological and /or physical tools (manipulatives)	-Before/after school tutorial program -Leveled rubrics -Increased intervention -Small groups -Change in pace -Extended time -Alternative assessments -Tiered activities/products -Color coded notes -Use of movements -Use various forms of technology **Extension**: See Project Lead the Way *General Teacher Resources* for Modifications for Advanced Students	-Change in pace -Alternative assessments -Accommodations as per IEP -Modifications as per IEP -Use graphic organizer to clarify mathematical functions for students with processing and organizing difficulties. -Constant review of CSE concepts to strengthen understanding of prior concepts for difficulties recalling facts. -Use self-regulations strategies for student to monitor and assess their thinking and performance for difficultly attending to task -Cooperative learning (small group, teaming, peer assisted tutoring) to foster communication and strengthen confidence -Use technology and/or hands on devices to: clarify abstract concepts and process for: 1. Difficulty interpreting pictures and diagram. 2. Difficulties with oral communications. 3. Difficulty correctly identifying Symbols. 4.Difficulty maintaining attentions -Simplify and reduces strategies/ goal structure to enhance motivation, foster independence and self-direction for: 1. difficulty attending to task 2. difficulty with following a sequence of steps to solution.	CSE journals Teacher Modeling Gestures Pictures/Photos Word Wall Native language Supports Partner Work **Extension**: See Project Lead the Way *Possible Student Misconceptions* for Modifications for English Language Learners.

		3. difficulty processing and organizing -Scaffolding CSE idea/concepts guided practice and questioning **Extension**: See Project Lead the Way *Teacher Resources* for Modifications for Advanced Students for Classroom Differentiation for Special Need Students. See Project Lead the Way *Possible Student Misconceptions* for Modifications for Classroom Differentiation for Special Need Students.	

IV. Instructional Resources and Materials

Formative Assessment	Summative Assessment	Supplemental Resources	Digital Resource
Teacher Observation Checks for understanding Exit tickets Unit Activities Unit Projects **Extension** – See activities and projects in above scope and sequence**.**	Unit Activities (PLTW) Unit Projects (PLTW)	**Teacher Resources** Project Lead the Way Code.org **Student Resources** Khan Academy code.org Project Lead the Way CSE program	Project Lead the Way Computer Science Essentials