SUBJECT: PLTW Innovators and Makers	GRADE: 7
Unit Title: Algorithms	**Time Frame: 8 Days**
UNIT OVERVIEW	
Within this unit students will learn about algorithms. By the end of this unit students will have made algorithms representing different actions, and flowcharts.	
LRG SKILLS AND DISPOSITIONS	**PA STANDARDS**
Critical Thinking and Problem Solving	**2-AP-10:** Use flowcharts and/or pseudocode to address complex problems as algorithms. **2-AP-13:** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
COMPETENCIES	**LEARNING TARGETS**
I can approach a challenge with computational thinking	• I can think in sequential steps. • I can break problems down into smaller steps. • I can explain an algorithm.

SUBJECT: PLTW Innovators and Makers	GRADE: 7
Unit Title: Learning the Microbit	**Time Frame: 12 Days**
UNIT OVERVIEW	
Within this unit students will learn about using the MicroBit. By the end of this unit students will have an understanding on how to use the MicroBit, and how to make a message appear on the MicroBit.	
LRG SKILLS AND DISPOSITIONS	**PA STANDARDS**
Creativity and Innovation	**1B-CS-02:** Model how computer hardware and software work together as a system to accomplish tasks. **2-CS-02:** Design projects that combine hardware and software components to collect and exchange data.

	2-CS-03: Systematically identify and fix problems with computing devices and their components. **2-AP-10:** Use flowcharts and/or pseudocode to address complex problems as algorithms. **2-AP-11:** Create clearly named variables that represent different data types and perform operations on their values. **2-AP-12:** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **2-AP-18:** Systematically test and refine programs using a range of test cases. **2-AP-19:** Document programs in order to make them easier to follow, test, and debug. **3A-AP-13:** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
COMPETENCIES	**LEARNING TARGETS**
I can code a program to express an idea or solve a problem.	• I can change and improve code that already exists.
I can approach a challenge with computational thinking.	• I can understand the purpose and outcome of a program. • I can think in sequential steps. • I can create programs by creating and testing codes in an incremental approach.
I can use best practices while programming.	• I can make sure my project meets the requirements. • I can create and follow a plan to solve a problem, using short-term and long-term goals.

SUBJECT: PLTW Innovators and Makers	**GRADE: 7**
Unit Title: Inputs and Outputs	**Time Frame: 9 Days**
UNIT OVERVIEW	
Within this unit students will learn about inputs and outputs. By the end of this unit students will be able to identify inputs and outputs along with their function and how to program them.	
LRG SKILLS AND DISPOSITIONS	**PA STANDARDS**
Continual Learning and Growth Mindset	**2-CS-02**: Design projects that combine hardware and software components to collect and exchange data. **2-NI-04**: Model the role of protocols in transmitting data across networks and the Internet.

	2-NI-06: Explain how physical security measures protect electronic information. **2-AP-11:** Create clearly named variables that represent different data types and perform operations on their values.
COMPETENCIES	**LEARNING TARGETS**
I can code a program to express an idea or solve a problem.	• I can use inputs, outputs and events to create an interactive program. • I can create and edit variables. • I can use a variety of variables to store and recall different values. • I can determine the value of variables at all points of the program • I can identify events that will occur when the program is running and write the code to respond to those events. • I can edit existing code to solve a new problem
I can approach a challenge with computational thinking.	• I can think in sequential steps. • I can understand and explain the code in the program • I can describe the purpose of a section of code.
I can demonstrate an understanding of many digital devices.	• I can identify and use inputs and outputs as tools. • I can create and use different types of data • I can identify and describe user-interface, data values, event handlers and procedures
I can use best practices while programming.	• I can use correct terminology • I can use pair programming techniques to complete a project • I can describe how working together can result in the best product • I can use good programming practices to make code more readable

SUBJECT: PLTW Innovators and Makers	**GRADE: 7**
Unit Title: Secrets and Safes Prototype	**Time Frame: 8 Days**
UNIT OVERVIEW	
Within this unit students will use what they have learned to create a prototype to keep something secret or safe. By the end of this unit students will have worked with a partner to create a prototype.	
LRG SKILLS AND DISPOSITIONS	**PA STANDARDS**

Collaboration and Teamwork	**2-CS-01:** Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. **2-CS-02:** Design projects that combine hardware and software components to collect and exchange data. **2-AP-16:** Seek and incorporate feedback from team members and users to refine a solution that meets user needs. **2-AP-18:** Systematically test and refine programs using a range of test cases.
COMPETENCIES	**LEARNING TARGETS**
I can code a program to express an idea or solve a problem.	● I can use inputs, outputs and events to create an interactive program. ● I can create my own ideas and build upon other projects ● I can use appropriate code to solve problems ● I can create and edit variables. ● I can create algorithms using variables, conditionals, operators and logic. ● I can identify events that will occur when the program is running and write the code to respond to those events.
I can approach a challenge with computational thinking.	● I can think in sequential steps. ● I can break problems down into smaller steps ● I can plan and create a project by breaking it into smaller parts using procedures. ● I can understand and explain the code in the program ● I can debug a program using a variety of methods ● I can use the iterative process to solve a problem ● I can create programs by creating and testing code in an incremental approach ● I can use computational thinking to solve problems. ● I can describe the purpose of a section of code. ● I can create a program using different planning techniques such as pseudocode or flowcharts
I can demonstrate an understanding of many digital devices.	● I can identify and use inputs and outputs as tools. ● I can create and use different types of data ● I can understand that computers process information quickly ● I can collect and interpret data to understand different problems ● I can identify and describe user-interface, data values, event handlers and procedures
I can use best practices while programming.	● I can use correct terminology

	• I can use pair programming techniques to complete a project • I can describe how working together can result in the best product • I can define how a team will function and measure success • I can test code frequently to assure that it is working correctly. • I can recognize that identifying and defining problems and proposing a solution can be challenging. • I can use good programming practices to make code more readable • I can make sure my project meets the requirements • I can create and follow a plan to solve a problem, using short-term and long-term goals.

SUBJECT: PLTW Innovators and Makers	**GRADE: 7**
Unit Title: Final Project Prototype	**Time Frame: 8 Days**
UNIT OVERVIEW	
Within this unit students will use what they have learned to create a prototype of their own. By the end of this unit students will have worked with a partner to create a prototype.	
LRG SKILLS AND DISPOSITIONS	**PA STANDARDS**
Collaboration and Teamwork	**2-CS-01:** Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. **2-CS-02:** Design projects that combine hardware and software components to collect and exchange data. **2-AP-15:** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts **2-AP-16:** Seek and incorporate feedback from team members and users to refine a solution that meets user needs. **2-AP-18:** Systematically test and refine programs using a range of test cases. **2-IC-20:** Discuss issues of bias and accessibility in the design of existing technologies.
COMPETENCIES	**LEARNING TARGETS**
I can code a program to express an idea or solve a problem.	• I can use inputs, outputs and events to create an interactive program. • I can create my own ideas and build upon other projects • I can use appropriate code to solve problems

	• I can create and edit variables. • I can create algorithms using variables, conditionals, operators and logic. • I can identify events that will occur when the program is running and write the code to respond to those events.
I can approach a challenge with computational thinking.	• I can think in sequential steps. • I can break problems down into smaller steps • I can plan and create a project by breaking it into smaller parts using procedures. • I can understand and explain the code in the program • I can debug a program using a variety of methods • I can use the iterative process to solve a problem • I can create programs by creating and testing code in an incremental approach • I can use computational thinking to solve problems. • I can describe the purpose of a section of code. • I can create a program using different planning techniques such as pseudocode or flowcharts
I can demonstrate an understanding of many digital devices.	• I can identify and use inputs and outputs as tools. • I can create and use different types of data • I can understand that computers process information quickly • I can collect and interpret data to understand different problems • I can identify and describe user-interface, data values, event handlers and procedures
I can use best practices while programming.	• I can use correct terminology • I can use pair programming techniques to complete a project • I can describe how working together can result in the best product • I can define how a team will function and measure success • I can test code frequently to assure that it is working correctly. • I can recognize that identifying and defining problems and proposing a solution can be challenging. • I can use good programming practices to make code more readable • I can make sure my project meets the requirements • I can create and follow a plan to solve a problem, using short-term and long-term goals.