A. CONTACTS	
1. School/District Information:	School/District: Chino Valley Unified School District Street Address: 5130 Riverside Dr. Chino, Ca. 91710 Phone: (909) 628-1201 Web Site: chino.k12.ca.us
2. Course Contact:	Teacher Contact: Office of Secondary Curriculum and Instruction Position/Title: Director of Secondary Curriculum and Instruction Site: District Office Phone: (909) 628-1201 X1630
B. COVER PAGE - COURSE ID	
1. Course Title:	Coding Connections
2. Transcript Title/Abbreviation:	Coding Connect
3. Transcript Course Code/Number:	3113
4. Subject Area/Category:	Computer Science
5. Grade Level(s):	8
6. Length of Course:	One Year
7. Classified as a Career Technical Education Course:	Yes
8. Date of Board Approval:	May 19, 2022

9. Course Description:

This course is designed for students interested in the computer science field but have limited technical knowledge of Minecraft Education, Minecraft MakeCode, and Python Coding. Students will learn the basics of coding using the Minecraft Education Platform. Students explore the foundational concepts related to coding while also being introduced to the K-12 Computer Science Standards and ISTE Standards. This course includes video lessons, reading units, practice performance tasks, performance assessments, and students will be introduced to different careers in the computer science field. As students perform various tasks in Minecraft Education, they will gain a broad understanding of coding which will allow them to better understand how it is used in their everyday life and help them refine their interest and make vocational choices.

10. Prerequisites:	None

11. Context for Course:

The goal for this course is to introduce students to various computer science principles in preparation to move for Chino High School's Biomedical Science and Technology Academy (BST).

12. History of Course Development:

This course has been designed as the entry level course that will introduce students to various skills within computer science. Students will be utilizing Minecraft Education Edition to work with the different coding techniques specifically focusing on Python Coding. The development of this course was for students to be exposed to coding and computer science so that they would be interested in moving forward to Chino High's BST Academy.

13. Textbooks (online Curriculum)	No textbooks required.
14. Supplemental Instructional Materials:	• Microsoft Education instructional materials. • Code.org
C. COURSE CONTENT	

1. Course Purpose:

This course is designed for students to be introduced to the basics of computer science and coding that will allow them to move forward with the BST Academy pathway at Chino High School. This course serves as an elective course that will allow students to work with Minecraft: Education Edition while also mastering various concepts when it comes to

coding. Students will receive basic level training that will prepare them for future courses in the BST Academy. This course will prepare them for different courses in the academy, Introduction to Python, Computer Sciences, etc. Students will be able to build a foundation in computer science which will promote their success as they move through the pathway at BST. This course will not only introduce students to coding but they will be exposed to the K12 Computer Science Standards and ISTE standards.

2. Course Outline:

Unit 1 - Introduction

This unit introduces Minecraft: Education Edition and Microsoft MakeCode. Students will learn about computer programming and change their Minecraft world (like make it rain chickens) all through coding.

Learning Goals:

- Describe computer science and coding and its importance

- Learn to play and maneuver in Minecraft: Education Edition

- Understand block programming in Microsoft MakeCode

- Change your Minecraft world through coding

K12 Computer Science Standards:

- **CPP.L1:6-05** Construct a program as a set of step-by-step instructions to be acted out.
- **CPP.L1:6-06** Implement problem solutions using a block-based visual programming language.

Unit 2 - Events

In this unit, students will learn about events and event handlers, which are important concepts in computer science and can be found in all programming languages. Students will start with a fun unplugged activity that demonstrates cause and effect, and how events trigger actions in the real word. This unit will provide students an opportunity to get hands-on with MakeCode in Minecraft, and finally, students will challenge themselves to create their own Make Code projects that use events to activate different parts of the program.

Learning Goals:

- Describe the different kinds of coding events

- Understand the importance of events while playing Minecraft

- Understand the importance of events in coding

- Learn about real-life events and how they can affect situations

- Alter the Minecraft landscape through coding with events

- Design an original creative project to change their Minecraft world through coding events

K12 Computer Science Standards:

- **CPP.L1:6-05** Construct a program as a set of step-by-step instructions to be acted out.
- **CPP.L1:6-06** Implement problem solutions using a block-based visual programming language.

Unit 3 - Coordinates

This unit covers how to move around Minecraft world with respect to the three-coordinate grids represented by (X, Y, Z) coordinates and the difference between relative position and world positions. The unit includes two unplugged activities to demonstrate coordinates and relative vs. world positions in real life. This unit will provide students with a hands-on experience with guided coding activities to create a compass rose, design a tool that allows students to copy and paste entire structures, and herd sheep in the Minecraft world. In the final activity students synthesize their learning with a self-directed, creative MakeCode coding project.

Learning Goals:

- Describe the difference between relative coordinates and absolute coordinates in real life

- Identify your real-world position and calculate the position of a landmark or object relative to your position

- Describe the difference between relative player position and absolute world position in Minecraft

- Understand the importance of coordinates in coding while playing Minecraft

- Alter the Minecraft landscape by coding with coordinates

- Design an original creative project to apply your coding skills in new ways

K12 Computer Science Standards:

- **3A-IC-26** Demonstrate ways a given algorithm applies to problems across disciplines.
- **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- **3B-AP-14** Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
- **CT.L2-12** Use abstraction to decompose a problem into sub problems.
- **CPP.L1:6-05** Construct a program as a set of step-by-step instructions to be acted out.
- **CPP.L1:6-06** Implement problem solutions using a block-based visual programming language.

Unit 4 - Variables

In this unit, students will explore the concept of a variable, an important way to store information and make programs more flexible and adaptable. This unit will build on what was learned about events in previous units, by using a variable to pass additional information when an event occurs. In this unit, students will be provided with opportunities to write their own program using variables to customize how it runs.

Learning Goals:

- Describe the different kinds of coding variables

- Understand the importance of variables while playing Minecraft

- Understand the importance of variables in coding

- Learn about real-life variables and how they can affect situations

- Alter the Minecraft landscape through coding with variables

- Design an original creative project to change their Minecraft world through coding an automated solution with variables

K12 Computer Science Standards:

- **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.
- **CT.L1:6-01** Understand and use the basic steps in algorithmic problem-solving.
- **CT.L1:6-02** Develop a simple understanding of an algorithm using computer-free exercises.
- **CPP.L1:6-05** Construct a program as a set of step-by-step instructions to be acted out.
- **2-A-5-7** Create variables that represent different types of data and manipulate their values.

Unit 5 – Conditionals

In this unit, students will explore the concept of conditionals. An important part of programming is telling the computer when to perform a certain task. Conditionals accomplish this by requiring a certain condition or rule to be met before an action is performed. Students will get the opportunity to code activities to create various games that use collaborative design and pair programming techniques to solve a problem players encounter in Minecraft by coding conditionals.

Learning Goals:

- Describe the importance of conditionals in coding

- Create IF THEN and IF THEN ELSE conditional statements

- Code with a variety of conditional blocks to automate their agent to find and collect important resources

- Evaluate code to identify problems like infinite loops and debug the code with conditionals

- Add a Say Block inside If Then Blocks to help debug problem code

- Work collaboratively to design an original creative project to apply their coding skills in new ways

K12 Computer Science Standards:

- **CL.L2-05** Implement problem solutions using a programming language, including looping behavior, conditional statements logic, expressions, variables, and functions.

- **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.L2-04** Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, socialization.

- **CL.L3A-01** Work in a team to design and develop a software artifact.

- **K-12 Computer Science Framework Core Concept:** Control Structures.

Unit 6 – Functions

In this unit students will explore the concept of a function. Often in programming, there are tasks or procedures that are used frequently within the same program. Rather than rewrite the lines of code that perform a particular task each time they need it, they can group the set of instructions together as a function. Grouping frequently used instructions as a function makes your code more efficient. Students can write the set of instructions once as a function and from then simply 'call' the function from inside your program whenever they need a task done. Students will then use functions and give the function a name that describes the task it will perform when called, making the code easier to read.

Learning Goals:

- Describe the different kinds of coding functions

- Understand the importance of functions while playing Minecraft

- Understand the importance of functions in coding

- Learn about real-life functions and how they can affect situations

- Alter the Minecraft landscape through coding Design an original creative project to change their Minecraft world through coding an automated solution with functions

K12 Computer Science Standards:

- **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

- **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse.

- **3A-CS-01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

- **3A-AP-13** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

Unit 7 – Iteration

In this unit, students will explore the concept if iteration, or ways to make things repeat. Programmers might repeat actions in a program to have a certain effect or might use repetition to accomplish the same task in a small number of steps. Students will get familiar with the concepts through various activities. This unit will provide opportunities to experience guided coding activities to get to know their agent. Students will take the knowledge they have learned with Minecraft and be challenged to incorporate iteration through various tasks.

Learning Goals:

- Understand the use of iteration in coding and synonymous terms used by programmers

- List examples of iteration in daily life

- Describe the different types of loops in MakeCode

- Use different types of loops to debug and code more efficiently

- Design an original creative project to automate a solution with iteration

K12 Computer Science Standards:

- **CL.L2-05** Implement problem solutions using a programming language, including looping behavior, conditional statements logic, expressions, variables, and functions.
- **CL.L3A-03** Explain how sequence, selection, iteration, and recursion are building blocks of algorithms.

Unit 8 – Arrays

In this unit, students will explore the concept of arrays. In computer science, an array is a series of places to store things. Earlier in this course, students learned how to use variables to store information. Arrays store multiple values, and they are accessed from a single object, the array itself. Students will continue the idea of storing information for later use, but for this unit, students will use many values. This unit will allow for students to learn the usefulness of arrays as a collection of related items. Students will learn how to use the array operations to add and get items to arrange and remember.

Learning Goals:

- Understand the usefulness of arrays in coding as a collection of related items

- List examples of arrays in real life

- Use the array operations to add and arrange items

- Describe array indexes and elements in MakeCode

- Sort array values with basic sorting algorithms

- Design an original creative project that uses arrays to create a piece of artwork in Minecraft

K12 Science Standards:

- **3A-DA-09** Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
- **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-14** Create procedures with parameters to organize code and make it easier to reuse.
- **K-12 Computer Science Framework Core Concept:** Control Structures.
- **CT.L2-12** Use abstraction to decompose a problem into sub problems.
- **CPP.L1:6-05** Construct a program as a set of step-by-step instructions to be acted out.
- **CPP.L1:6-06** Implement problem solutions using a block-based visual programming language.
- **NGSS 3-5-ETS1-2** Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

Unit 9 – Artificial Intelligence

In this unit, students will explore the concept of artificial intelligence (AI). The science of crafting programs that mimic, and even surpass human intelligence, is tremendously important. However, there are also some ethical questions, and fears when it comes to artificial intelligence. Students will explore these ideas through various discussions and assignments. Students will be given an opportunity to create an intelligent agent in Minecraft, they will code for their agent to build intelligently and adapt to the Minecraft environment.

Learning Goals:
- Identify factors that distinguish humans from machines
- Recognize that computers model intelligent behavior
- Understand the importance of artificial intelligence and explore some of the ethics and fears relating to artificial intelligence
- Find opportunities to code your agent to behave intelligently in Minecraft
- Design an original creative project to teach your agent to intelligently adapt to the Minecraft environment

K12 Computer Science Standards:
- **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.L2-04** Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding, and accepting multiple perspectives, socialization.
- **CL.L3A-01** Work in a team to design and develop a software artifact.
- **K-12 Computer Science Framework Core Concept:** Control Structures.
- **CT.L2-12** Use abstraction to decompose a problem into sub problems.
- **CPP.L1:6-05** Construct a program as a set of step-by-step instructions to be acted out.
- **CPP.L1:6-06** Implement problem solutions using a block-based visual programming language.
- **NGSS 3-5-ETS1-2** Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

Lesson 10 – Final Project
In this final unit, students will explore the final chapter of MakeCode for Minecraft, students will create a final project that does two things:
1. Show what they know
2. Demonstrate something new

Learning Goals:
- Design an original creative project to program four tools for a survival backpack to help them in a Minecraft world.
- Demonstrate their learned coding skills and apply them in a new way.
- Validate their approach to the project, including beta testing and analysis of code to debug and problem solve.

K12 Computer Science Standards:
- This unit and final project is set to incorporate all standards that have been incorporated into this course.

3. Key Terms, Key Assignments, and Projects

Unit 1 – Introduction
Students will be introduced to what computer science is and the various jobs that are available to them in the computer science field. Students will select various careers to see what they would be able to do outside in the real-world by researching careers. Since this unit is an introduction, students will be given simple tasks in Minecraft to ensure they know the tools and keys of Minecraft. Makecode will be introduced to students where they will complete basic coding assignments to provide a foundation before moving forward into the next units.

Unit 2 – Events
Students will be introduced to events through different assignments and be able to demonstrate their understanding of the learning goals for this unit. There will be 4 major assignments:
- Introduction to events
- Coding with events
- Linking events

Project/Assessment:
<u>Getting creative with events</u>
Students will use their skills at the end of this unit to demonstrate the mastery of events through an end of unit project using multiple handler event blocks to change their Minecraft world.

Unit 3 – Coordinates
Students will complete assignments for each lesson in this unit to demonstrate their understanding of coordinates. The lessons will focus on four main concepts within this unit:
- Introduction to coordinates
- Coding with coordinates
- Automating actions with coordinates

Project/Assessment:
<u>Get creative with coordinates</u>
Students will use their skills at the end of this unit to demonstrate the mastery by designing an original creative project to apply their coding skills to Alter the Minecraft landscape by coding with coordinates.

Unit 4 – Variables
Students will complete assignments for each lesson in this unit to demonstrate their understanding of variables. The lessons will focus on four main concepts within this unit and students will be given an opportunity to show their mastery of the topic:
- Introduction to variables
- Coding with variables
- Combining variables

Project/Assessment:
<u>Get creative with variables</u>
Students will use their skills at the end of this unit to Design an original creative project to change their Minecraft world through coding an automated solution with variables.

Unit 5 – Conditionals
Students will complete assignments for each lesson in this unit to demonstrate their understanding of conditionals. The lessons will focus on four main concepts within this unit and students will be given an opportunity to show their mastery of the topic:
- Introduction to conditionals
- Coding with conditionals
- Debug problem code with conditionals

Project/Assessment:
<u>Get creative with conditionals</u>
Students will use their skills at the end of this unit to demonstrate the mastery of conditionals by collaboratively working to design an original creative project to apply their coding skills in new ways.

Unit 6 - Functions
Students will complete assignments for each lesson in this unit to demonstrate their understanding of functions. The lessons will focus on four main concepts within this unit and students will be given an opportunity to show their mastery of the topic:
- Introduction to functions
- Coding with functions
- Building on functions

Project/Assessment:

Get creative with functions

Students will use their skills at the end of this unit to alter the Minecraft landscape through coding by designing an original creative project to change their Minecraft world through coding an automated solution with functions.

Unit 7 – Iteration

Students will complete assignments for each lesson in this unit to demonstrate their understanding of iteration. The lessons will focus on four main concepts within this unit and students will be given an opportunity to show their mastery of the topic:

- Introduction to iteration
- Coding with iteration
- Debugging with iteration

Project/Assessment:

Get creative with iteration

Students will use their skills at the end of this unit to design an original creative project to automate a solution with iteration.

Unit 8 – Arrays

Students will complete assignments for each lesson in this unit to demonstrate their understanding of arrays. The lessons will focus on four main concepts within this unit and students will be given an opportunity to show their mastery of the topic:

- Introduction to arrays
- Coding with arrays
- Building on arrays

Project/Assessment:

Get creative with arrays

Students will use their skills at the end of this unit to design an original creative project that uses arrays to create a piece of artwork in Minecraft.

Unit 9 – Artificial Intelligence

Students will complete assignments for each lesson in this unit to demonstrate their understanding of artificial intelligence. The lessons will focus on two main concepts within this unit and students will be given an opportunity to show their mastery of the topic:

- Introduction to artificial intelligence
- Coding an intelligent agent

Project/Assessment:

Get creative with artificial intelligence

Students will use their skills at the end of this unit to design an original creative project to teach your agent to intelligently adapt to the Minecraft environment.

Unit 10 – Final Project

Students will complete a final project that will allow them to demonstrate their understanding of all the topics that were taught during this course. The project will encompass all major topics and require students to utilize all information learned in this course to complete a final project. Students will also use their skills demonstrate the mastery of all standards by designing an original creative project to program four tools for a survival backpack to help them in a Minecraft world. Students will validate their approach to the project, including beta testing and analysis of code to debug and problem solve. The project will be divided into three parts:

- Introduction to project

- Coding the project
- Beta test and finalize the project

4. Instructional Methods and/or Strategies:
- This course uses an online curriculum which gives the students access to video lessons and virtual simulator exercises to complete lab projects and assessments.
- Students are expected to complete the virtual labs and practice questions assessments from each unit.
- Each topic will be discussed during class with the students.
- Students will be introduced to vocabulary that will continue to build upon one another in the course.
- Students will complete hands-on projects that reflect the material learned in the online curriculum.

Other strategies would include:
- Direct instruction
- Small group work
- Activity-based instruction
- Collaborative groups
- Modeling
- Group discussions

5. Assessment Including Methods and/or Tools:

The evaluation of student progress will be based on the following criteria outlined in board policy:
- Assessments: 60% of the final grade
- Assignments and class discussions: 40% of the final grade