Educator Toolkit

COMPUTER SCIENCE EDUCATOR GUIDE

Introduction

Are you curious how you might begin computer science in your classroom? Are you looking for a unique way to engage your students into learning new skills and content? With technology playing an increasingly important role in every profession, a foundational understanding of computer science is becoming an essential component of student learning. The need for authentic and engaging computer science instruction is more important than ever.

Minecraft: Education Edition is pleased to introduce the K-12 Computer Science Progression. Within the progression, you will find a computer science content collection with over 150 hours of high-quality computer science materials intended for all students in their educational journey.

The material has been exclusively designed to afford teachers a consistent and easy-to-implement curriculum resource. Teachers can leverage the novelty and engagement of the Minecraft worlds to help students to acquire content knowledge and develop relevant skills. By using Minecraft as a platform for learning, educators, like you, can motivate and inspire every student to achieve more and ignite a passion for learning.

So how can you get started? You don't need to be an expert Minecraft player to start teaching with Minecraft: Education Edition! In fact, Minecraft: Education Edition is a great place to start your computer science journey. Make sure you have the necessary hardware and software

Materials

Hardware	- The teacher will need a laptop or tablet with a projector for the plugged-in coding lessons. - Internet access will be required for a portion of the lessons and activities. - Each student will need a device to complete the plugged-in coding activities. - OPTIONAL: Some educators prefer students to use headphones during the independent work segment of the lessons, as there is noise within the game.
Software	- Minecraft: Education Edition needs to be deployed on the devices utilized within these lessons. Use this link to find information about Deploying Minecraft: Education Edition. - The teacher and students will need the relevant MCworld files downloaded for each lesson.
Other Materials	- Certain lessons will call for basic supplies such as copies of handouts, pencils, markers, etc. Each individual lesson will specify if any additional materials are needed outside of the hardware and software.

Getting Started with Minecraft: Education Edition

Students

There are no required prerequisites around computer science, coding, or tech skills for students. Students should have had some instruction and/or guidance around device usage and navigation (i.e., appropriate use of hardware, device login procedure/information, digital safety, etc.). There is a recommended grade level/age range for students as it relates to the developmentally-appropriateness and reading comprehension levels of the computer science units. Please refer to the Computer Science Progression on page 4.

Teachers

Teachers do not need to have any prior computer science experience; however, they should familiarize themselves with a basic understanding of Minecraft: Education Edition. Support for building out teacher knowledge can be found here:

Minecraft: Education Edition: Teacher Academy

Minecraft: Education Edition Webinar Series

Install Minecraft: Education Edition

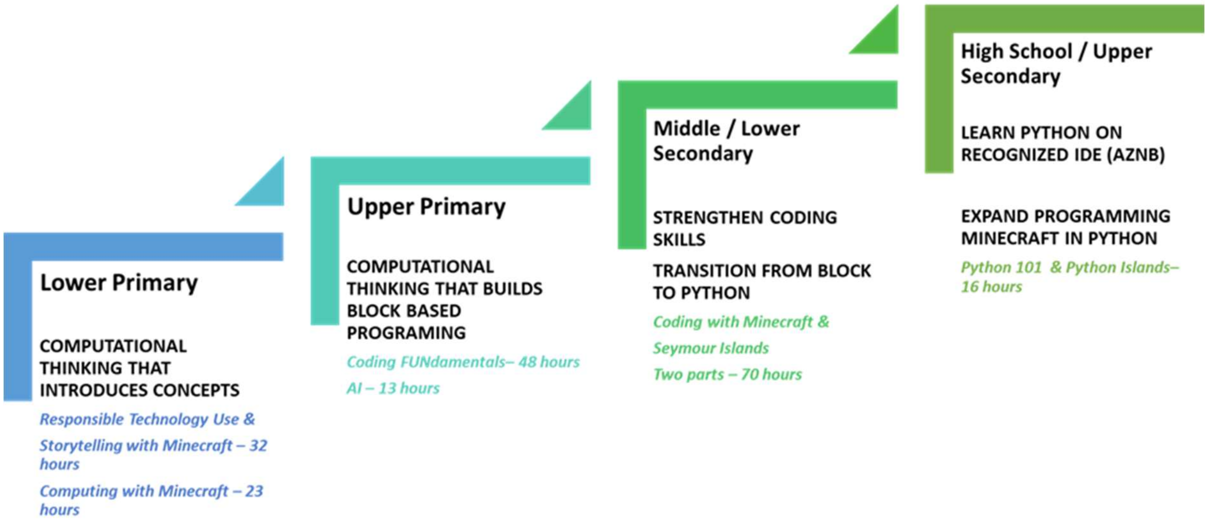
Minecraft: Education Edition can be installed on Chromebook, iPad, Mac, and PC. To ensure your experience with Minecraft: Education Edition is top-notch, make sure your devices meet the minimum system requirements. To check if your device supports Minecraft: Education Edition, see System Requirements.

If you have questions about setting up Minecraft: Education Edition, the following link will provide you with some frequently asked questions and additional information about set-up. On this page, you will find assistance for:

- Get Started
- Purchase Licenses
- Administration and License Management
- Installation
- Troubleshooting

K-12 Computer Science Progression

The Minecraft computer science kit is specially designed computer science content to address relevant academic standards in a logical, sequential progression. Students will learn, practice, and apply relevant computer science skills and concepts in both unplugged and digital experiences. The lessons are designed to follow the gradual lease methodology (refer to the Lesson Design section of this guide found below). Students should have multiple opportunities to practice and demonstrate mastery in any given skill/concept.



Lower Primary

COMPUTATIONAL THINKING THAT INTRODUCES CONCEPTS

Responsible Technology Use & Storytelling with Minecraft – 32 hours

Computing with Minecraft – 23 hours

Upper Primary

COMPUTATIONAL THINKING THAT BUILDS BLOCK BASED PROGRAMING

Coding FUNdamentals– 48 hours

AI – 13 hours

Middle / Lower Secondary

STRENGTHEN CODING SKILLS

TRANSITION FROM BLOCK TO PYTHON

Coding with Minecraft &

Seymour Islands

Two parts – 70 hours

High School / Upper Secondary

LEARN PYTHON ON RECOGNIZED IDE (AZNB)

EXPAND PROGRAMMING MINECRAFT IN PYTHON

Python 101 & Python Islands– 16 hours

In order to provide teachers with a consistent and easy-to-follow format, the instructional materials for the Minecraft computer science pathway is standardized. All units within the entire progression will follow the same instructional methodology and include all of the same resources to ensure a consistent and high-quality experience for both teachers and students.

Unit Materials

Curriculum Overview	This computer science unit document will provide you with a complete overview of the content, skills, and learning outcomes intended to be taught. It will include the theme and/or focus of the unit, the relevant grade levels or age group, connection to the Minecraft Teaching and Learning framework, essential question, objectives, academic standards, suggest lessons and pacing, assessments, and glossary.
Educator's Guides (Lesson Plan)	The educator's guide provides a theme overview, learning goals, standards addressed, required preparation for the activities, key vocabulary, the lesson plans for the activities, and any additional materials needed.
Classroom Presentations	Each lesson (educator's guide) is supported by its own PowerPoint presentation to provide structure and guide the educator through the activities for the lesson. This will support the lesson delivery for students.
Formative Assessments	These assessments are intended to provide a quick check for understanding based on the concepts and skills taught during the specific lesson. Each formative assessment is 2-4 questions and should be completed at the end of the lesson. These are found within the educator's guides.
Summative Assessment	This is a culminating experience intended to assess students' learning of all the computer science concepts taught within the unit in an authentic and meaningful way.
Minecraft World Files	These are the specific world files needed to experience the instructional activities that have been linked directly within the Educator's Guides. If there is no file listed within the lesson plan, then it is an open build. Students should use a Blocks of Grass template (or any other appropriate template) to complete the lesson.

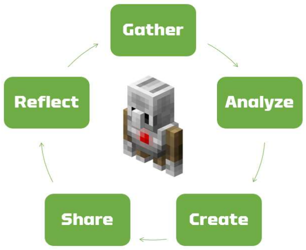
Lesson Design

Within the Minecraft Computer Science K-12 Progression, there are various curriculum units for students. Within each unit, there are multiple lessons (which are known as the Educator's guides). All lesson activities within the Educator's Guide will contain three parts correlating with the gradual release model:

Direct Instruction—"I Do"	In the first step, the teacher models the appropriate way of performing the skills included in the new concept being taught.
Guided Instruction—"We Do"	After the teacher models the correct way to understand or perform the new concept being taught, they partner with the students and work through some examples together (partner with teacher and/or classmates) .
Independent Practice—"You Do"	This step is where students demonstrate their initial level of understanding of the new concept being taught through independent practice.

Minecraft: Education Edition Teaching and Learning Framework

<table>
<tr><th colspan="2">GOAL:
Provide students with the necessary skills to address the complex societal issues of our time and their future</th></tr>
<tr><td><b>Coding Mindset</b>
Although Minecraft: Education Edition provides rigorous and engaging academics, we also deliver a holistic education- providing instruction for the social, emotional, and physical needs for our students. We want to foster a distinctive set of attributes. These qualities prepare our students to make exceptional contributions both in school and outside of school.
CURIOUS
COMMITTED
COOPERATIVE
CONSIDERATE
CONFIDENT</td><td><b>Computational Thinking Skills</b>
Computational thinking provides a vital skill set in which students must possess in order to fulfill the industry's needs in the jobs of tomorrow.
Our ever-changing workforce creates a critical need for innovation. Our students need computational thinking skills not just to solve the problems within their educational journey, but to also meet the challenges of adapting to our constantly changing workforce.
DECOMPOSITION
PATTERN RECOGNITION
ABSTRACTION
ALGORITHMS</td></tr>
</table>

Computer Science Units of Study	**Community**
Minecraft: Education Edition provides meaningful, relevant, and engaging units of study. The units of study will possess a conceptual lens to allow for depth and complexity to develop conceptual understanding—knowledge which transfers through time, across cultures, and across situations. **DIGITAL CITIZENSHIP** **PROGRAMMING** **CYBERSECURITY** **IMPACTS OF COMPUTING**	As students gain and possess new knowledge and skills, we strive for them to find a greater purpose of "why do I need to know this" or more importantly, "how can I use this information?". We aim to empower students develop confidence and self-efficacy into a commitment to serve the community in which we live in and beyond. 

This framework provides a larger perspective of how to support robust and comprehensive computer science in the classroom. There are four core elements, which summarize the overall experience:

- Coding Mindset: It is essential we teach and provide opportunities for students to exercise their social and emotional capacities as it relates to learning. Our goal is to instill a coding mindset in every student who works within our continuum of learning. We want students to be curious, committed, cooperative, considerate, and ultimately confident.
- Computational Thinking: The Minecraft computer science experience affords students with the opportunities to develop their computational thinking skills. These skills will provide a construct of how students can think critically and problem solve.
- Computer Science Units of Study: This is the learning content centered around supporting mastery of the computer science standards. It will provide meaningful and relevant learning experiences for students to engage in computer science.
- Community: As students gain and possess new knowledge and skills, we strive for them to find a greater purpose of "why do I need to know this" or more importantly, "how can I use this information?". Computer science can help us to solve the problems of today and in the future. We encourage students to use their newfound knowledge and skills for the greater good.

Computer Science Standards

This next section will provide you with an overview of the full computer science curriculum and its corresponding academic standards. The curriculum sequence is presented in chronological order—we suggest working in order within the grade level bands, as the content will build upon skills presented in the previous unit. However, educators should feel empowered to modify and adapt the content and/or sequence to best meet the needs of their students.

Grade Level	Curriculum Unit Name	Number of Lessons	Hours	CSTA Standards Addressed
K-2	Responsible Technology Use	6 lessons	10 hours	1A-CS-02 1A-CS-03 1A-NI-04 1A-AP-11 1A-DA-06 1A-DA-07 1A-IC-17 1A-IC-18
K-2	Storytelling in Minecraft	12 lessons	22 hours	1A-IC-17 1A-AP-12
K-2	Computing with Minecraft	16 lessons	21 hours	1A-AP-10 1A-AP-11 1A-AP-12 1A-AP-14 1A-AP-15 1A-AP-17 1A-AP-18
3-5	Coding Fundamentals (Part 1)	10 lessons	16 hours	1B-AP-08 1B-AP-10 1B-AP-11 1B-AP-15 1B-AP-17
3-5	Coding Fundamentals (Part 2)	10 lessons	16 hours	1B-AP-08 1B-AP-10 1B-AP-11 1B-AP-15 1B-AP-17
3-5	Coding Fundamentals (Part 3)	6 lessons	12 hours	1B-AP-13 1B-AP-16 1B-AP-08 1B-AP-10

				1B-AP-11 1B-AP-15 1B-AP-17
3-5	Artificial Intelligence	8 lessons	13 hours	1B-IC-18 1B-IC-20 1B-AP-08 1B-AP-11 1B-AP-15 1B-AP-17
6-8	Coding with Minecraft (Part 1)	17 lessons	21 hours	2-DA-08 2-AP-10 2-AP-11 2-AP-12 2-AP-13 2-AP-14 2-AP-18 2-AP-19
6-8	Coding with Minecraft (Part 2)	21 lessons	40 hours	2-AP-17 2-DA-08 2-AP-10 2-AP-11 2-AP-12 2-AP-13 2-AP-14 2-AP-18 2-AP-19
6-8	Seymour Island	1 lesson	3 hours	2-AP-11 2-AP-12 2-AP-14
9-12	Python 101 Python Islands Python Playgrounds	17 lessons	24 hours	3A-AP-13 3A-AP-14 3A-AP-15 3A-AP-16 3A-AP-17 3A-AP-18 3A-AP-21 3A-AP-22 3A-AP-23

Computer Science Concepts

This next section will provide you with an overview of the full computer science curriculum and the CS concepts covered within the unit.

Grade Level	Curriculum Unit Name	Number of Lessons	Hours	CS Concepts Addressed
K-2	Responsible Technology Use	6 lessons	10 hours	Algorithms Devices Data Computing Systems Troubleshooting Impacts of Computing Social Interactions Cybersecurity Safety
K-2	Storytelling in Minecraft	11 lessons	22 hours	Sequencing Algorithms/Instructions/Commands Digital Communication Digital Content Decomposition Intellectual Property Attribution Collaboration Digital Artifacts Debugging Collaboration Program Development Testing
K-2	Computing with Minecraft	16 lessons	21 hours	Sequencing Algorithms Decomposition Variables Coordinates/Positions Loops Functions Abstraction Debugging Collaboration Program Development Testing

3-5	Coding Fundamentals (Part 1)	10 lessons	16 hours	Credibility/Accuracy of Resources Public Domain/Creative Commons Copyright Intellectual Property Decomposition Sequencing Control Flow (structure) Events + Event Handlers Algorithms Pattern Recognition Loops (repeat loop, nested loop, conditional loop) Conditionals Debugging Program Development Collaboration
3-5	Coding Fundamentals (Part 2)	10 lessons	16 hours	Credibility/Accuracy of Resources Public Domain/Creative Commons Copyright Intellectual Property Decomposition Sequencing Control Flow (structure) Events + Event Handlers Algorithms Pattern Recognition Loops (repeat loop, nested loop, conditional loop) Conditionals Debugging Program Development Collaboration
3-5	Coding Fundamentals (Part 3)	6 lessons	12 hours	Decomposition Sequencing Control Flow (structure) Events + Event Handlers Algorithms Pattern Recognition Loops (repeat loop, nested loop, conditional loop) Conditionals Debugging Program Development

				Collaboration
3-5	Artificial Intelligence	8 lessons	13 hours	Impacts of Computing Feedback Decomposition Data Data Visualization Algorithms Debugging Program Development Iteration Computational Artifacts
6-8	Coding with Minecraft (Part 1)	17 lessons	21 hours	Decomposition Data Collection Algorithms Events Event Handlers Documentation Control Flow (structure) Loops Variables Conditionals Debugging Program Development
6-8	Coding with Minecraft (Part 2)	21 lessons	40 hours	Decomposition Algorithms Control Flow (structure) Functions Iteration Debugging Arrays Artificial Intelligence Machine Learning Program Development
6-8	Seymour Island	1 lesson	3 hours	Decomposition Sequencing Nested Loops Compound Conditionals Algorithms Debugging Program Development Cryptography
9-12	Python 101	17 lessons	24 hours	Python commands Syntax structure

	Python Islands Python Playgrounds			String (text) Integers (numbers) Coordinates Positions Variables Lists Zero based numbering For loops Nested loops Conditionals (if | if else | elif) Boolean logic While loops Functions Decomposition Outputs Logic operators Debugging Using Reference Guide