

Nuner Innovation Lab, students will gradually take on more responsibility as inventors, designers, coders, mathematicians, problem-solvers, and creators:

- **August:** Innovation Lab Boot Camp with Hour of Code
- **September:** Pattern Programmer Training with CodeSpark
- **October:** Creative Coding Lab with Scratch Basics
- **November–December:** Design Lab Challenges
- **January–March:** Math Arcade Studio — Multiplication Innovation Projects
- **April:** Quality Assurance, Testing, and Revision
- **May:** Math Arcade Expo, where students host another class, families, teachers, or administrators

Yearlong Coding-to-Math Plan: Nuner Innovation Lab

yearlong progression that builds confidence first, develops computational thinking next, and then moves students into original game design connected to multiplication fluency.

Hour of Code → CodeSpark → Scratch → Math Arcade Studio.

Yearlong Progression Overview

Tool or Phase	Purpose	Time Needed
Hour of Code	Inspire students, introduce coding concepts, and help students believe “I can do this.”	First 2 weeks
CodeSpark	Build computational thinking through structured puzzles, sequencing, loops, debugging, and perseverance.	4–6 weeks
Scratch	Teach students to create animations, simple games, variables, events, movement, conditionals, and player feedback.	October–December
Math Arcade Studio	Apply coding skills to student-created multiplication games with authentic playtesting, revision, and presentation.	January–May

Classroom Theme: Nuner Innovation Lab

Instead of presenting coding as a separate subject or limiting the work to video games, I’ll frame the entire year as membership in **Nuner Innovation Lab**. Students are inventors, designers, coders, mathematicians, problem-solvers, and creators who use technology to build learning experiences for others.

My Launch message: “Welcome to Nuner Innovation Lab. This year, you are not just students, you are inventors, designers, coders, mathematicians, and problem-solvers. Your mission is to create digital projects that help others learn. Every innovator starts in training, and over the next two weeks, you will begin earning badges toward becoming an Innovation Lab Designer.”

8 Badges to earn throughout the year:

- **Level 1: Coding Explorer:** Earned during Hour of Code and Coding Boot Camp when students complete an unplugged coding activity, explain what an algorithm is, try an Hour of Code challenge, and show perseverance when something does not work the first time. Student-friendly statement: “I can explore coding, follow step-by-step instructions, and keep trying when I get stuck.”
- **Level 2: Pattern Problem Solver:** Earned during CodeSpark Training Camp when students complete sequencing challenges, identify a loop or repeated pattern, explain how they fixed a bug, and connect loops to

skip counting or multiplication patterns. Student-friendly statement: “I can use patterns, loops, and debugging to solve coding puzzles.”

- **Level 3: Creative Coder:** Earned during Scratch Basics when students create a project with a sprite, use an event, add movement, speech, animation, or sound, and explain what their project is supposed to do. Student-friendly statement: “I can use coding blocks to create something original.”
- **Level 4: Design Thinker:** Earned during Design Lab Challenges when students plan an interactive project, add a goal or user interaction, use partner feedback, and explain how a user will experience their design. Student-friendly statement: “I can plan, build, test, and improve a project for an audience.”
- **Level 5: Logic Leader:** Earned during Scratch skill-building when students use variables, conditionals, random numbers, scorekeeping, or answer checking and debug a logic error. Student-friendly statement: “I can make my program think, decide, and respond.”
- **Level 6: Math Innovator:** Earned during Math Arcade Studio when students create a project where multiplication is central, use accurate factors and products, and explain how the math and coding work together. Student-friendly statement: “I can use coding to help others learn math.”
- **Level 7: Quality Tester:** Earned during the April revision phase when students test their own project, playtest a partner’s project, find and fix at least one bug, and revise based on feedback. Student-friendly statement: “I can test, debug, and improve my work like a real innovator.”
- **Level 8: Expo Creator:** Earned during the Math Arcade Expo when students prepare their final project for visitors, share a designer statement, explain how their project helps others learn multiplication, and participate in the showcase. Student-friendly statement: “I can share my finished project with an audience and explain my learning.”

Weekly Structure: Two 30-Minute Sessions

Each week follows the same rhythm so coding stays manageable within the math schedule while still giving students enough time to learn, build, debug, and reflect.

- **Session 1: Studio Lesson.** Introduce one coding or design concept, connect it to math or game design, model the skill, and have students try a small challenge.
- **Session 2: Studio Build Time.** Students continue building, apply the concept, peer test, debug, and complete a quick reflection.

August: Innovation Lab Boot Camp with Hour of Code

Purpose: Build excitement, confidence, and shared vocabulary before students begin CodeSpark or Scratch. The goal is not mastery yet; the goal is for students to believe coding is possible, fun, and worth trying.

Week 1: What Is Coding?

- **Session 1: Studio Lesson.** Introduce algorithms as step-by-step instructions. Use an unplugged activity such as giving precise directions to draw a simple picture or complete a classroom routine.
- **Session 2: Studio Build Time.** Complete one Hour of Code activity. End by discussing what students noticed about instructions, mistakes, persistence, and problem solving.

Week 2: Think Like a Programmer

- **Session 1: Studio Lesson.** Introduce vocabulary: algorithm, sequence, bug, debug, loop, event, and perseverance. Connect each word to something students experienced during Hour of Code.
- **Session 2: Studio Build Time.** Complete a second Hour of Code activity, celebrate progress, and award the Level 1: Coding Explorer badge.

September: Pattern Programmer Training with CodeSpark

Purpose: Use CodeSpark to strengthen computational thinking before students move into open-ended creation. CodeSpark keeps the cognitive load low because students are solving puzzles while quietly learning sequencing, loops, conditionals, debugging, and persistence.

- **Weeks 3–4:** Sequencing and precise instructions. Students complete puzzles that require actions in the correct order and explain why order matters.
- **Weeks 5–6:** Loops and repeated patterns. Students identify repeated actions and connect loops to skip counting and multiplication patterns.
- **Weeks 7–8:** Debugging and persistence. Students practice finding mistakes, revising plans, and explaining how they fixed a problem.

Badge connection: Students earn the Level 2: Pattern Problem Solver badge when they can explain a sequence, identify a loop, connect repeated patterns to math, and describe one bug they fixed.

October: Creative Coding Lab with Scratch Basics

Purpose: Transition from guided puzzle solving to open-ended creation. Students learn the Scratch interface slowly, so they do not become overwhelmed before the multiplication game work begins.

- **Week 9:** Sprites, backdrops, events, and movement. Students create a sprite that moves and speaks when the green flag is clicked.
- **Week 10:** Looks, costumes, and animation. Students create a short animation connected to a math vocabulary word or multiplication fact.
- **Week 11:** Loops and movement patterns. Students create repeated movement that connects to multiples or skip counting.
- **Week 12:** Simple interaction. Students use an ask block, a response, or a click event to make the project interactive.

Badge connection: Students earn the Level 3: Creative Coder badge when they create a short Scratch project that includes an event, a sprite action, and a clear purpose for the viewer or user.

November–December: Design Lab Challenges

Purpose: Teach students how interactive projects work before asking them to design multiplication learning experiences. During this phase, students build small design challenges with scoring, movement, feedback, timers, win screens, simple playtesting routines, and user experience decisions.

- **Week 13:** What makes a game? Students compare a plain quiz with a themed game and identify goals, rules, feedback, challenge, and player motivation.
- **Week 14:** Scorekeeping. Students create a score variable and make the score increase after a correct action.
- **Week 15:** Conditionals. Students use if/then or if/else logic to make the game respond differently depending on the player's choice.
- **Week 16:** Randomness. Students use random numbers to make a game feel different each time it is played.
- **Week 17:** Playtesting. Students test a simple game, notice bugs, and revise one feature based on feedback.
- **Week 18:** Design Challenge 1. Students create a small arcade-style game that includes a goal, score, instructions, and feedback.

Badge connection: Students earn the Level 4: Design Thinker badge when they can plan, build, test, and improve an interactive project for an audience.

January–March: Math Arcade Studio — Multiplication Innovation Projects

Purpose: This is where coding and math fully merge. Students use Scratch to design original math learning experiences that strengthen multiplication fluency. Projects may be games, quizzes, escape rooms, interactive stories, puzzles, digital flashcards, boss battles, racing challenges, or simulations. The coding skills should now be familiar enough that students can focus on math accuracy, user experience, and creative problem solving.

January: Multiplication Game Mechanics

- **Week 19:** Build a multiplication quiz engine using factors, products, ask blocks, and answer checking.
- **Week 20:** Add scorekeeping, correct/incorrect feedback, and a reset at the start of the game.
- **Week 21:** Add random factors so the game is replayable and supports mixed fact practice.
- **Week 22:** Add a win condition such as reaching 10 points, escaping a room, crossing a finish line, or defeating a boss.

Badge connection: Students earn the Level 5: Logic Leader badge when they add scorekeeping, feedback, and at least one conditional or random element to a playable project.

February: Studio Teams and Game Themes

- **Week 23:** Students form studio teams or work individually, choose a theme, and write a one-sentence game mission.
- **Week 24:** Students storyboard the player experience: start screen, challenge, feedback, win condition, and ending screen.
- **Week 25:** Students build the first playable version of the game, also called Version 1.0.
- **Week 26:** Students add polish such as directions, backdrops, sounds, costumes, or animations while keeping multiplication central.

March: Expansion and Differentiation

- **Week 27:** Students add difficulty levels, such as facts 1–5, facts 1–9, and facts 1–12.
- **Week 28:** Students add a second game mechanic such as movement, unlocking doors, racing, collecting items, or boss battles.
- **Week 29:** Students improve player feedback, so the game teaches, not just scores.
- **Week 30:** Students prepare for peer testing by checking math accuracy, instructions, scoring, and win conditions.

Badge connection: Students earn the Level 6: Math Innovator badge when they make their game bigger and better by adding new challenges, creating different difficulty levels, helping players learn from their mistakes, and getting their game ready for others to play and test.

April: Quality Assurance, Testing, and Revision

Purpose: Students learn that real innovators test, revise, and improve. Bugs are treated as information, not failure, and feedback becomes part of the design process.

- **Week 31:** Self-test using a checklist for math accuracy, directions, scoring, and game flow.
- **Week 32:** Peer playtest with feedback stems such as “I noticed...” and “I wonder...”
- **Week 33:** Revision sprint. Students fix bugs, improve instructions, adjust difficulty, and strengthen multiplication accuracy.
- **Week 34:** Final quality check. Students explain how their code supports multiplication learning.

Badge connection: Students earn the Level 7: Quality Tester badge when they thoroughly test their game, use feedback to make meaningful improvements, fix bugs, and explain how their final design and code help players practice multiplication accurately and effectively.

May: Math Arcade Expo

Purpose: Give students an authentic audience for their work. The Innovation Lab opens its final showcase as students host a Math Arcade Expo where another class, families, teachers, or administrators can explore their projects and give feedback.

- **Week 35:** Prepare arcade stations, game directions, designer statements, and voting categories.
- **Week 36:** Host the Math Arcade Expo. Visitors play games and vote on categories such as Most Creative, Best Gameplay, Best Math Learning Experience, Hardest Challenge, and Students' Choice.
- **Week 37:** Reflect on the design process. Students explain what they built, what they revised, and how their game helps players practice multiplication. .

Badge connection: Students earn the Level 8: Expo Creator badge when they share their game with others, teach players how to play, explain how their game helps people practice multiplication, and celebrate all the hard work they put into creating, testing, and improving their game.

Indiana Grade 5 Mathematics Standards Integrated

Standard	Connection to Innovation Lab
5.CA.2 – Solve real-world problems involving multiplication and division of whole numbers.	Students design multiplication games that require accurate multiplication facts, scoring systems, and authentic problem solving while creating playable games.
5.CA.1 – Use multiplication and division strategies and explain reasoning.	Students continually apply multiplication concepts when designing quiz engines, answer checking, and game logic while explaining how their code works.
Mathematical Process Standards (embedded throughout <i>Indiana Math</i>)	Students model, reason, persevere, communicate mathematical thinking, and revise solutions while debugging programs and testing games. These practices are embedded in every phase of the Innovation Lab.
5.DA.1 – Collect, organize, and analyze data.	During playtesting and the Math Arcade Expo, students collect user feedback, observe gameplay, and use the results to improve their games.
5.DA.2 – Analyze data to make decisions.	Students evaluate testing results and peer feedback to revise game difficulty, instructions, scoring, and player experience.

Indiana Grade 5 Science / Technology (Engineering Design) Standards Integrated

Standard	Connection to Innovation Lab
3-5-ETS1-1 – Define a design problem with criteria and constraints.	Students identify the goal of creating a multiplication game that teaches math while meeting specific design expectations such as accuracy, gameplay, and user experience.
3-5-ETS1-2 – Generate and compare multiple solutions.	Students brainstorm game themes, mechanics, and layouts before selecting and refining the strongest design.
3-5-ETS1-3 – Test and improve prototypes.	Students self-test, peer test, identify bugs, revise, improve, and optimize their games before presenting them at the Math Arcade Expo.
Science & Engineering Practice: Developing and Using Models	Students use storyboards, Scratch projects, algorithms, and prototypes as models throughout the design process.
Science & Engineering Practice: Using Mathematics and Computational Thinking	Students apply computational thinking, variables, conditionals, loops, debugging, algorithms, and multiplication concepts to solve authentic design challenges.

Standard

Science & Engineering Practice:
Constructing Explanations and
Designing Solutions

Science & Engineering Practice:
Obtaining, Evaluating, and
Communicating Information

Connection to Innovation Lab

Students explain coding decisions, justify revisions, and communicate how their games help others learn multiplication.

Students present their projects during the Math Arcade Expo, explain their design process, and respond to audience questions and feedback.

3D-Printed Badges: Throughout the year, students will learn how to design, prepare, and program files for our 3D printers as they help create the badges they earn in coding. Each printed badge celebrates a new milestone while giving students hands-on experience with 3D printing technology and the engineering design process.



