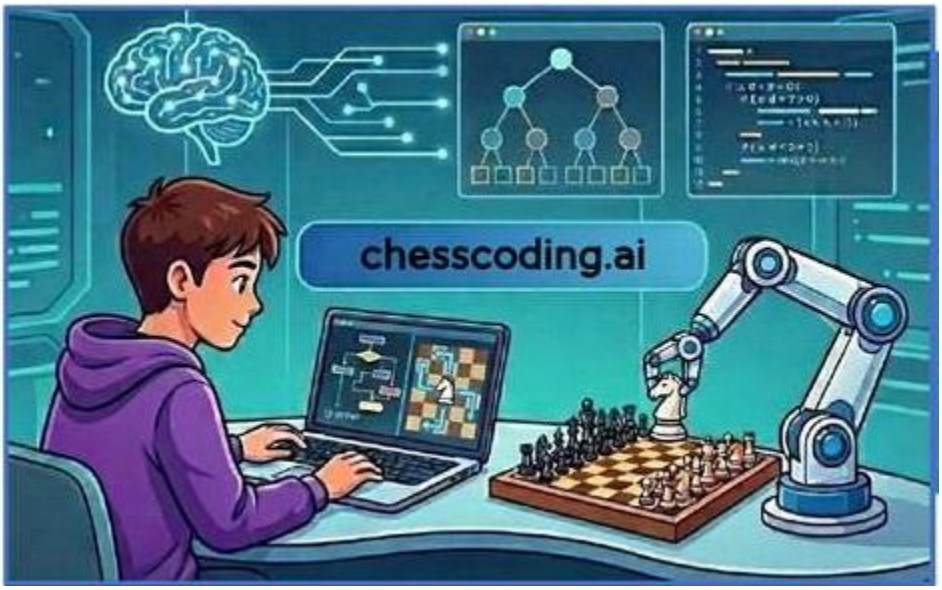
CHESSCODING.AI




# Parent & Student Value Guide

# What Should Families Consider When Comparing AI and Coding Summer and After-School Programs?

When families compare AI and coding programs—whether as a summer bootcamp, weekend course, or after-school program—it is helpful to look beyond the campus name, the number of advertised activities, or the total length of the day.

Programs with similar prices can provide very different educational experiences. Important questions include:

- How much of the schedule is devoted to actual instruction?
- How large is the class?
- Who teaches the program?
- Will students write and modify code themselves?
- Does the curriculum teach transferable concepts?
- Is the program built around a coherent learning progression?
- What will students understand, build, and retain afterward?

# Understanding the General Price Range

Based on a review of 2026 technology and pre-college programs, one-week commuter coding and technology programs commonly fall within the approximate range of **$850–$1,500**.

Two-week programs hosted at or near university campuses often cost approximately:

- **$3,000–$4,500** for commuter participation
- **$5,000–$6,500 or more** when housing and meals are included

Programs designed for younger primary-grade students are sometimes less expensive than programs offering advanced coding, artificial intelligence, engineering, or pre-college content.

These are general market ranges rather than exact comparisons. Prices vary according to location, program length, housing, equipment, class size, instructor qualifications, and the amount of direct instruction provided.

# Scheduled Time Is Not Always Instructional Time

A program may be advertised as a full-day experience while including only three to five hours of lessons and supervised project work.

The rest of the day may include:

- Lunch and breaks

- Recreation and games
- Campus tours
- Career-development seminars
- Guest speakers
- Social activities
- Excursions
- General camp programming

These activities can be enjoyable and valuable. However, they are different from direct instruction, supervised coding, debugging, problem-solving, and individual academic feedback.

Families should therefore ask not only, “How long is the day?” but also:

> How many hours will the student spend actively learning, coding, solving problems, and receiving guidance?

ChessCoding.ai provides a minimum of **30 hours of guided instruction and supervised coding over five days**.

At the standard tuition of $900, this represents approximately **$30 per guided instructional hour**. More importantly, those hours are focused on computer science, artificial intelligence, coding exercises, algorithmic thinking, and supervised project development.

# Why Class Size Matters

Many programs do not publicly state their student-to-instructor ratio.

A class of 10–15 students may work well for lectures and demonstrations. However, coding education often requires individual attention. Students encounter different errors, interpret instructions differently, and progress at different speeds.

When a student’s program does not run correctly, the instructor may need to:

- Inspect the student’s code
- Identify a logical or syntax error
- Ask questions about the student’s reasoning
- Explain the concept in a different way
- Provide an easier intermediate step
- Offer a more advanced challenge when the student is ready

ChessCoding.ai is designed around a small-group model of approximately **four students per instructor**.

This allows time for individual feedback, questions, debugging, discussion, and adjustment to different levels of coding and chess experience.

Personalized instruction is especially important when students are working with advanced concepts such as recursion, search algorithms, dynamic programming, game trees, and reinforcement learning.

## The Instructor Matters

A strong curriculum is important, but the instructor determines how effectively that curriculum reaches the student.

Some programs clearly identify their instructors and provide information about their academic, professional, or teaching backgrounds. Other programs advertise a broad collection of camps without identifying the person who will teach a particular group.

Both models can produce positive experiences. However, families may wish to ask:

- Is the instructor clearly identified?
- Does the instructor have expertise in the subject?
- Has the instructor worked successfully with young learners?
- Can the instructor explain complex concepts in accessible language?
- Can the instructor support both beginners and advanced students?
- Will the instructor personally teach the course or primarily supervise prepared activities?

ChessCoding.ai is led by an experienced educator with a background in competitive chess, computer science, youth education, and artificial intelligence curriculum development.

The instructor has worked closely with high-performing young chess students and understands the importance of individualized instruction. This combination of technical knowledge, competitive experience, and teaching practice helps make difficult concepts approachable while allowing advanced students to explore them more deeply.

The focus is not only on delivering information. It is also on mentoring students, encouraging curiosity, strengthening independent thinking, and helping each participant develop confidence in their ability to solve problems and write code.

## Curriculum Rigor, Depth, and Breadth

Curriculum rigor is another important factor when assessing the quality and value of an AI or coding program. Programs with similar schedules and prices may differ substantially in how deeply they explore a subject. Some provide an enjoyable introduction through short demonstrations, guided activities, or a single platform, while others expect students to write code, solve challenging problems, compare algorithms, debug their work, and understand the ideas behind the technology.

ChessCoding.ai is designed to offer greater depth and breadth than a typical introductory technology camp. Rather than focusing narrowly on one game, coding tool, robotics kit, or AI application, the curriculum follows a coherent progression from Python and computational thinking to recursion, search algorithms, minimax, dynamic programming, reinforcement learning, Monte Carlo Tree Search, and AlphaZero-style systems. These topics are connected through historical puzzles, mathematical problems, guided implementation, experimentation, and discussion.

The program's structure is informed by the instructor's firsthand experience as an MIT alumnus with the rigor and sustained effort expected in demanding university courses. Our weeklong, full-day summer programs and multi-weekend school-year programs attempt to create, at an age-appropriate level, the continuity and progression of a complete course rather than a collection of unrelated activities. ChessCoding.ai is therefore intended for families seeking substantial guided learning, broad exposure to computer science and AI, and opportunities for students to investigate selected topics in meaningful depth.

# About the Location and MIT Faculty Support

Families should distinguish between a program operated directly by a university and an independent educational program that receives faculty or departmental support to use university teaching space.

ChessCoding.ai is an independent educational initiative run by an MIT alumnus. It has received support from several MIT faculty members for the educational program and for the use of MIT campus teaching space. Under MIT's event and facilities procedures, an outside organization generally requires an MIT faculty or staff sponsor, together with the applicable departmental approval, to organize an educational program on campus.

In this administrative sense, the program may receive MIT faculty or departmental sponsorship for the use of teaching space. This sponsorship reflects support for hosting the educational activity and compliance with the relevant campus procedures. It should not, however, be interpreted as meaning that ChessCoding.ai is operated by MIT or that MIT institutionally endorses every aspect of the program, unless such endorsement is separately and explicitly authorized.

When a session is held in an MIT facility, it may therefore be accurately described as an independently organized program held on the MIT campus with the required faculty or departmental sponsorship for space use. When a session is held at a nearby facility, it should be described as being held near or next to the MIT campus.

At present, some sessions may be organized near the MIT campus to reduce facility, insurance, and administrative costs while maintaining convenient access to Cambridge's academic and technology environment. An MIT campus tour may also be included as part of the student

experience. This arrangement allows more of the program's resources to be directed toward instruction, curriculum, and small-group mentorship.

# MIT Campus Tour

Our program includes a half-day educational tour of the MIT campus. Campus visits are included in many technology, coding, and AI summer programs, but their educational depth can vary considerably. Some tours are primarily designed for sightseeing, photographs, and general exposure to a prestigious university environment.

The ChessCoding.ai campus tour is designed as an extension of the curriculum. Led by an MIT alumnus familiar with the Institute's educational culture and its K–12 resources, the tour helps students connect the AI, mathematics, engineering, research, and entrepreneurship concepts discussed in class with real academic pathways and institutions.

Students and parents are introduced to examples of MIT-related opportunities and resources, such as MIT RAISE, Beaver Works, MIT PRIMES, MITES, the Edgerton Center, and the Martin Trust Center for MIT Entrepreneurship. Where appropriate, we also explain which opportunities are available to K–12 students, which are highly selective, and which become relevant only after admission to MIT.

The goal is not simply to show students the campus. It is to help them understand how a young learner can progress from curiosity and coding practice to advanced projects, research opportunities, university study, technological innovation, and entrepreneurship.

The campus tour also provides context about MIT's academic culture: hands-on experimentation, interdisciplinary problem-solving, collaboration, responsible innovation, and the practical application of scientific knowledge. This makes the visit more closely connected to the student's learning journey than a standard campus walk.

## Future STEM, Research, and Entrepreneurship Opportunities

ChessCoding.ai helps students and parents become aware of advanced academic opportunities they may explore as students progress through middle school, high school, and eventually college.

**Selective STEM and Research Programs for High School Students**

- **MITES Summer** — A free, six-week residential STEM program for high school juniors, particularly students from underserved or underrepresented communities.
- **MITES Semester** — A free, six-month hybrid STEM and college-preparation program for high school juniors.
- **Research Science Institute (RSI)** — A highly selective, free six-week program combining advanced STEM coursework with hands-on scientific research.

- **Beaver Works Summer Institute (BWSI)** — An intensive four-week program offering advanced subjects such as autonomous systems, cybersecurity, quantum software, engineering, and artificial intelligence.
- **Women's Technology Program (WTP)** — An intensive engineering and computer science program designed to expand participation among students historically underrepresented in these fields.
- √**MathROOTS** — A free, two-week residential mathematics program for nationally selected high-potential students.
- **Summer Science Program (SSP)** — A residential research program through which students investigate fields such as astrophysics, biochemistry, genomics, and synthetic chemistry.
- **MIT PRIMES** — A free, year-long research program for mathematically talented high school sophomores and juniors. Students work with academic mentors on unsolved problems in mathematics, theoretical computer science, and computational biology. PRIMES also includes programs for students outside Greater Boston and younger students.

**Hands-On Science and AI Education**

- **MIT Edgerton Center High-Speed Imaging Programs** — The Edgerton Center continues Professor Harold "Doc" Edgerton's pioneering work in high-speed photography. Its educational activities introduce students to strobe photography, high-speed video, engineering experimentation, and the scientific study of motion. The center's **Extreme Imaging** summer program allows rising ninth- through twelfth-grade students to explore high-speed, microscopic, ultraviolet, infrared, and astrophotography.
- **MIT RAISE — Responsible AI for Social Empowerment and Education** — MIT RAISE develops hands-on, age-appropriate artificial-intelligence curricula and educational programs for K–12 students and teachers. Its initiatives include the **Day of AI**, responsible-AI curricula, teacher resources, and programs that encourage students not only to use AI, but also to understand, build, question, and shape AI systems.

**Opportunities After Admission to MIT**

- **MIT Mathematics RAMP** — RAMP is an intensive summer mathematics program for incoming MIT first-year students who are interested in mathematics. It helps newly admitted students prepare for advanced university mathematics and provides an "on-ramp" to the MIT mathematics major. Unlike the high-school programs above, students become eligible only after being admitted to and choosing to attend MIT.

**MIT's Entrepreneurship and Innovation Ecosystem**

MIT students are surrounded not only by advanced science and engineering opportunities, but also by a strong culture of invention and entrepreneurship.

The **Martin Trust Center for MIT Entrepreneurship** teaches innovation-driven entrepreneurship through courses, practical frameworks, advising, co-curricular programs, facilities, and connections with experienced entrepreneurs. Its mission is to help MIT students transform technological and scientific ideas into products, organizations, and ventures that can create real-world impact.

ChessCoding.ai introduces students to this broader pathway:

**curiosity → mathematical and computational foundations → AI projects → research → innovation → entrepreneurship**

We want students to understand that learning Python and artificial-intelligence algorithms can eventually lead not only to stronger academic performance, but also to research projects, scientific discoveries, new products, startups, and socially valuable technologies.

These programs are independently operated and generally have their own eligibility, application, and selection requirements. ChessCoding.ai is not affiliated with these programs and cannot guarantee admission. Our purpose is to help families discover advanced opportunities early and encourage students to develop the coding ability, mathematical reasoning, project experience, intellectual curiosity, and long-term ambition required to pursue them.

# A Transferable, Technology-Independent Curriculum

ChessCoding.ai uses an **award-winning, open-access youth AI curriculum** developed through an international educational initiative.

The curriculum is not built around a particular commercial computer game such as Roblox or Minecraft. It is also not dependent on one robotics kit, electronics platform, graphics card, commercial AI product, or proprietary hardware system.

Programs centered on commercial games or specialized equipment can be highly engaging. However, students may sometimes learn primarily how to operate that particular platform.

ChessCoding.ai takes a different approach.

Students study transferable computer science and artificial intelligence concepts that can be applied across many technologies, industries, and future areas of study.

The open-access nature of the curriculum also makes its educational approach more transparent. Families are not paying merely for access to hidden content. The principal value of the bootcamp comes from:

- Expert teaching
- Structured progression
- Guided coding

- Small-group mentorship
- Individual feedback
- Adaptation to the student's level
- Supervised problem-solving
- Discussion and reflection
- Access to learning materials that students can continue exploring

# Recovering the History of Computer Science and AI

An important part of the curriculum is the history of computer science and artificial intelligence—an area that often receives limited attention in modern courses focused mainly on programming syntax, current software tools, or the newest AI platforms.

Students benefit from understanding that today's technologies did not appear suddenly. They developed through centuries of mathematical puzzles, logical games, algorithmic ideas, and attempts to model human reasoning and decision-making.

The Shatranj.ai curriculum helps address this gap by connecting historical problems such as:

- The Wheat and Chessboard problem
- The Knight's Tour
- The Eight Queens puzzle
- Medieval chess and shatranj studies
- Classical problems of logic, search, and calculation

These historical problems are connected to modern computer science and AI concepts including:

- Recursion
- Backtracking
- Graph search
- Optimization
- Dynamic programming
- Game-tree analysis
- Reinforcement learning
- Self-learning systems

The curriculum also follows the development of computer chess from early theoretical ideas and experimental programs to Deep Blue, Stockfish, and AlphaZero.

This historical progression helps students understand not only **how** an algorithm works, but also:

- Why it was developed
- What problem it was intended to solve

- What limitations earlier methods encountered
- How newer approaches improved upon previous ones
- How mathematics, games, computing, and artificial intelligence developed together

History is therefore not presented as an optional collection of facts. It becomes a teaching framework that helps students understand the development of computational ideas.

# What Students Explore

The curriculum introduces students to areas including:

- Python programming and computational thinking
- Mathematical and historical board-game problems
- Board and game-state representation
- Recursion and backtracking
- Breadth-first and depth-first search
- Heuristic and graph-search algorithms
- Minimax and alpha-beta pruning
- Dynamic programming and tablebases
- Chess engines and computer decision-making
- Reinforcement learning and Q-learning
- Monte Carlo Tree Search
- AlphaZero-style self-learning systems

Chess and historical board games serve as understandable laboratories for studying these ideas.

Students can see the problem, represent it in code, test different approaches, compare results, and observe how an algorithm makes decisions.

The purpose is not simply to teach students how to play chess. It is also not merely to show them how to use a particular AI product.

Instead, students investigate questions such as:

- How can a computer represent a problem?
- How does an algorithm search through possible solutions?
- How can unnecessary possibilities be eliminated?
- How does a machine evaluate competing choices?
- How can previous results be stored and reused?
- How can an intelligent system learn from experience?
- How can different algorithms be compared?

These ideas also apply to robotics, logistics, navigation, game development, optimization, planning, economics, engineering, and intelligent decision-making.

## Appropriate Challenge and Realistic Expectations

Not every student is expected to completely master every advanced algorithm in a single week.

The program is designed to provide students with:

- A meaningful map of computer science and AI
- Substantial hands-on coding experience
- Exposure to both classical and modern algorithms
- Experience solving increasingly difficult problems
- Greater confidence in computational thinking
- A foundation for continued study
- A clearer understanding of possible interests in AI, computer science, engineering, mathematics, or research

Some students may leave with a strong understanding of Python fundamentals and search algorithms. Others may progress further into reinforcement learning, chess-engine analysis, or independent projects.

The small-group format allows the instructor to adjust the depth and pace according to the students participating in each session.

## Questions Families and Students Should Ask

When evaluating any AI or coding program, families and students may wish to ask:

- How many hours involve genuine instruction and supervised project work?
- How much of the day is devoted to recreation, tours, or general activities?
- What is the student-to-instructor ratio?
- Is the instructor's identity and experience clearly presented?
- Will students write and modify code themselves?
- Will students receive individual help when they become stuck?
- Is the curriculum tied to one commercial product, game, or hardware platform?
- Are the concepts transferable to future technologies and fields of study?
- Does the curriculum follow a coherent progression?
- How intellectually challenging is the content?
- What code, projects, recordings, or learning materials will students retain?
- Will students leave with a clearer understanding of how artificial intelligence actually works?

## The Value of ChessCoding.ai

The value of ChessCoding.ai comes from the combination of:

- An award-winning, open-access youth AI curriculum
- At least 30 hours of guided instruction and supervised coding
- An approximately 4:1 student-to-instructor ratio
- An experienced instructor in chess, computer science, and youth education
- Real programming and algorithmic problem-solving
- A technology-independent curriculum
- Historical and cultural context
- Individual mentorship and feedback
- A coherent progression from foundational problems to modern AI

The program is intended to help students move beyond being users of games, devices, and AI tools. The larger goal is to help them begin thinking like builders, problem-solvers, researchers, and future AI architects.

Ultimately, the purpose of this guide is not to suggest that one program format is right for every student, but to help families evaluate educational value more carefully. Price, location, and brand recognition matter, but they should be considered alongside instructional time, class size, instructor expertise, curriculum quality, intellectual challenge, and the amount of individual support a student will receive. ChessCoding.ai is designed for families seeking a focused, small-group experience that combines substantial guided coding, transferable computer science and AI concepts, historical perspective, and personal mentorship. Our aim is for students to leave not only with new code and technical knowledge, but also with greater confidence, curiosity, and a clearer understanding of how they can grow from users of technology into thoughtful builders, problem-solvers, researchers, and future AI architects.