$\mathbb{A}^2$ Summer Session Curriculum

Corey Predella & Gunner Peterson

Abstract Academy

ΔΣ Program Design Goals

We offer a transformative experience for our students. Specifically, our work is committed to a few of our goals as educators.

Foremost, our curriculum is carefully-crafted and original. We design 75% of our problems and exercises with the intent of the student generating the conceptual information using their own intuition and ideas. Increased intuition comes not from a 2-hour lecture, but instead, personalized and guided problem-solving, where each answer provides a missing puzzle piece to a larger picture. We also believe in exposing our students to classical problems in both mathematics and computer science, which is why the remaining 25% of our problems are sourced from national exams and competitions such as the SAT, ACT, AMC12, and AIME.

Our second commitment regards the larger goals of our program. Abstract Academy is not merely about incubating top-performers in the classroom. Instead, we are committed to instilling genuine curiosity and sustainable learning habits that will help students achieve stressless success in their learning endeavors. We understand that the U.S. school system imposes a belief on students that academic success comes from boring, exhaustive rote repetition, and we wish to combat this national crisis. We believe in teaching with variation because by exposing students to the same concept from two different perspectives, their understanding becomes 3D.

Finally, we have a ”no one left behind” policy. Only through persistent engagement will students grow into confident learners and problem solvers. It is our goal to foster an environment in which asking all questions (regardless of triviality) is within a safe and supported space. We will consistently ask students to share their own ideas and contribute to the learning environment so the multitude of perspectives contributes to a learning experience greater than the sum of its parts.

ΔΣ Week #1

By the end of the first week, students will have a strong basis of abstract mathematical/computational thinking along with a mastery of Python syntax excluding object oriented programming. A strong emphasis is placed on the mastery of basics in the first days in order to scaffold the thinking involved in complex topics during the rest of the program. For some days, the exact same topics from the math lesson may appear in the programming lesson. We believe that teaching with variation empowers intuition. Exposing students to the same concept from two different perspectives forces their understanding becomes 3D.

ΔΣ Day #1: Creative Problem Solving Mindset & Simple Programs

- **Morning Math Lesson (10:00AM - 12:00PM):** During The first hour of the morning lesson, we will discuss effective problem solving strategies. We will look into how to break harder problems into easier tasks and talk about how to gain intuition, which is the most powerful tool for creative thinking and problem solving. During the second hour, we will explore a compendium of mathematical puzzles and use the new techniques in action. We incorporate problems involving variables, substitutions, and binary numbers so that students are primed for the evening coding lesson.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program. Reference codes are included.

- **Evening Programming Lesson (4:00PM-6:00PM):** The first coding lesson will introduce variables and operations, which are the most fundamental tools of computer programming that scaffold ideas for all the later concepts. During the first hour, we prompt the students with hands-on examples that invent the ideas behind variables. In the second hour, we discuss operators and how operators combined with variables paired with automation form the foundation of computer programming.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program. Reference codes are included.
- **Game Night!**

By the end of day #1:

Skills: students will be comfortable with mathematical and computational substitution. Students will also be able to store variables using math and code, which is a handy technique to solve complex problems.
Goals: students will find the beginning concepts of computer programming intuitive after seeing them in action mathematically. They will also be able to take their substitution skills beyond the program, as substitution is common in Algebra 2+ and tested on the SAT and ACT.

$\Delta\Sigma$ Day #2: Unleashing Critical Thinking & Unlocking Logic

- **Morning Math Lesson (10:00AM - 12:00PM):** Since the evening coding lesson introduces conditional logic and statements, the morning math lesson is designed around casework which will build a strong logical intuition. We will introduce helpful counting techniques in the first hour along with convenient notation. Then, in the second hour, we will solve problems using new notation.
- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.
- **Evening Programming Lesson (4:00PM-6:00PM):** This coding lesson introduces conditional logic. We will show how to leverage conditional logic to make smart programs that adapt based on input variables. In the first hour, we will introduce binary statements and discuss decision trees through interactive examples. In the second hour, we will talk about logic gates and what happens when we attach the end of one gate to the beginning of another.
- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #2:

Skills: Students will be able to leverage casework in their problem solving ability, allowing them to use constraints to reach solutions. They will also be able to incorporate logical decision trees into their computer programs.
Goals: students will be able to locate and use logic in their daily lives through conversation and writing, which are two mediums where logic is a powerful tool.

$\Delta\Sigma$ Day #3: Noticing Patterns & Formatting Funky Functions

- **Morning Math Lesson(10:00AM - 12:00PM):** The morning lesson introduces the idea of functions, which is a key topic in most Algebra 2+ classes. But if one understands functions in math, then functions in code become a piece of cake. During the first hour, we will introduce functions and multivariate functions along with special variations. Through various examples, students will build intuition. Then, in the second hour, students will learn about function composition and functional recursion, which are two concepts that exist directly in computer programming.
- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** During the evening coding lesson, we will continue our studies of functions on the computer. We will show how to design effective functions and discuss the difference between functions with and without return values. In the first hour, we will walk through some examples in Python and discuss the differences in programmatic behavior. Then, in the second hour, we will tackle some coding problems in order to build independence.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Game Night!**

By the end of day #3:

Skills: Students will intuitively understand the ins and outs of functions. This intuition is applicable for both math and computer science as the topics refer to the same idea in different contexts. Students will also begin to ask questions about relationships between topics, allowing them to construct a web of mathematical and computational knowledge, which is a powerful learning tool to commit information to memory.

Goals: Students will be able to take their understanding of functions beyond the seminar. They will be able to understand Algebra 2+ intuitively because functions build the foundation for more complex units in math. Furthermore, function questions make up about 30% of eat SAT math section, meaning it is common in standardized tests.

$\Delta\Sigma$ Day #4: Exploring Higher Dimensions & Leveraging Loops

- **Morning Math Lesson (10:00AM - 12:00PM):** During the day #4 math lesson, students will be explore continuous sums and products and learn how to represent such mathematical operations with sigma/pi notation. Ultimately, the idea of sigma notation will build intuition when it comes to for loops and iteration in computer programming. During the first hour, we will prompt students with some mathematical questions involving incalculable sums using standard methods. We will ask students to create a creative way to solve these problems, and then generalize the solutions to problems involving infinity in the second hour.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** In the afternoon coding lesson, we will introduce looping systems, and how to leverage iterators to solve complex problems. During the first hour, we will talk about for loops and while loops while introducing the idea of iteration. In the second hour, we will talk about some algorithmic thinking techniques through interactive examples and group problems.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #4:

Skills: Students will understand the importance of defining notation during problem solving. Students will also be able to leverage iteration, which is the absolute most important concept for writing advanced algorithms.

Goals: By learning about sigma and pi notation, students will start to use abstraction in problem solving. They will begin to notice critical patterns that allow for simplification, which is crucial for breaking complex tasks into smaller ones.

ΔΣ Day #5: Putting Together Puzzle Pieces & Advanced Topics

- **Morning Math Lesson (10:00AM - 12:00PM):** The morning math lesson will focus on set theory, which is the most foundational and important theory for advanced mathematics. Students can use sets to categorize both mathematical objects and computational objects. During the first hour, we lead with a lecture on sets, and how certain restrictions can create categories. In the second hour, we will leverage set intuition to solve a series of fun combinatorial problems and puzzles.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** In the evening coding lesson, students will learn about generators, which is a special and powerful feature of the python programming language similar to set building. During the first hour, we will play a game of ”Generate the list” to encourage students to become comfortable with this technique. In the second hour, we will explore Lambda functions, which is another advanced technique in Python.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #5:

Skills: Students will have a complete knowledge of Python syntax. This means that if students can think of code intuitively, they can make any computer program they'd like to. Students will also build on their understanding of logic while exploring set theory, which will allow them to consider ideas in categories.

Goals: students will be able to think of an idea for a small-scale computer program and create it with 100% confidence that their syntax and ideas will work. Beyond the seminar, students will be able to feel confident in their learning and presenting abilities.

ΔΣ Week #2

talk about how students can now code in python but the rest is building the intuition

During the second week, students will dive into the world of complex and creative problem-solving through making algorithms and studying advanced topics in math. Students will see the symbiosis between number theory and coding, allowing them to leverage abstraction and critical thinking to generate essential and efficient computer programs. The thinking tools explored during this week are applicable to many contexts outside of computer programming in math. The ability to think critically is extremely valuable for high-level reading and writing.

ΔΣ Day #6: Revolutionizing Remainders & Introductory Algorithms

- **Morning Math Lesson (10:00AM - 12:00PM):**
  In the first lesson of week #2, we will enter into the world of complex problem solving. Specifically, we will discuss problems using modular arithmetic which is a technique that involved studying the remainder of numbers. Surprisingly, modular arithmetic – while it seems arbitrary – is an industry standard technique that offers a consistent way to optimize algorithms. During the first hour, students will follow a lecture on modular arithmetic and its applications with exploratory problems and critical discussions. Then, students will explore the world of number theory problems in the second hour, with each problem picked with the intent of scaffolding the ideas in the evening.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** In the evening, students will learn about runtime complexity, which is the standard technique used to measure how efficient a program is. Learning about runtime complexity is crucial for beginning programmers because it allows them to analyze their computational efficiency and maintain good habits while coding. During the first hour, we start off with a lecture on how to analyze computer programs including hands-on examples. In the second hour, students will use their new knowledge in a game of ”minimizer” in which students must create the most optimal codes to certain programming questions.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #6:
Skills: Students will be able to evaluate their computer programs for efficiency. Even though two different computer scripts can achieve the same results, students will be able to detect which program is more effective.
Goals: students will see the importance of evaluating their work, and how that process can lead to better results. They will be able to take this skill beyond the classroom whether that be for completing homework or studying for tests.

- **Game Night!**

ΔΣ Day #7: Inequalities & Maximizing Minimums

- **Morning Math Lesson (10:00AM - 12:00PM):** During the morning math lesson, students will learn about inequalities, which is a common unit in Algebra 2+. Students will learn to derive advanced inequalities using algebraic manipulation, giving them a set of tools to optimize their programs in the evening. During the first hour, we will go over basic examples of inequalities and increase complexity through the second hour.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** Students will learn about their first formal algorithm, called peak-finding. The reason students typically learn this algorithm first is because it can be adapted to many other algorithms such as trend-finding. In the first hour, students will explore the definition of a ”peak” and then be prompted to write a program to identify ”peaks” in a given list. In the second hour, student will reflect on their problem-solving process and use their knowledge to generalize to other algorithms.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #7:
Skills: Students will be able to reference a couple of different industry-standard algorithms from their tool belt. They will also start to have ”coding vision” in which they will see the correct way of solving a problem merely by reading the prompt.
Goals: students will start to ask the question ”have I seen a similar problem before?” which is one of the most effective problem solving strategies used today. They will also learn to view topics not exclusively, but instead as a set of building blocks waiting to be combined to solve a complex problem.

ΔΣ Day #8: Advanced Counting & Sorting Galore

- **Morning Math Lesson (10:00AM - 12:00PM):** The morning math lesson of day #8 is designed around advanced combinatorial techniques. Being able to quickly count how many ways there are to park 3 cars in a 6-car parking-lot allows students to generalize their technique to count how many ways there are to park n cars in a k-car garage. When it comes to programming, they can use their optimized formula in their algorithms. In the first hour, students will explore combinatorics and pascals triangle along with its many identities. Then, during the second hour, students will explore a series of meticulously-selected combinatorial puzzles

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** During the evening coding lesson, students will explore the world of sorting. They will consider their own sorting techniques in terms of its steps in an attempt to create an algorithm that can sort a list. This task is often viewed as the most crucial algorithm in computer programming as it can be adapted to many different forms. In the first hour, students will explore small cases of sorting. Then, they will generalize their techniques to sort lists of length n during the second hour.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Game Night!**

By the end of day #8:

Skills: Students will be able to count complicated scenarios and generalize problems into formulas using mathematical induction. Students will also be able to write a program to sort lists which is a crucial building block in many other algorithms.

Goals: Students will be learn to consider multiple instances of a mathematical property and create their own formulas. Students will be able to take this ability and succeed in math and science as they will be able to anticipate the content they learn, which is a common learning technique used to boost comprehension.

ΔΣ Day #9: Advanced Problem Solving Techniques & More Sorting

- **Morning Math Lesson (10:00AM - 12:00PM):** In the morning math lesson, students will reflect on the past two weeks of problem-solving and think about what techniques have worked, and what haven't. Then, we will introduce a set of seemingly-challenging problems. Before students give the problems a try, we will go over some final advanced problem solving techniques inspired by the world-renowned math book, *How To Solve It* by George Pólya. In the first hour, we will look into students' problem solving approach and discuss potential refinements. In the second hour, we will teach the "four phases" of solving a problem introduced by George Pólya and apply the method on some complex puzzles.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Evening Programming Lesson (4:00PM-6:00PM):** In the evening coding lesson, student will be prompted to create more optimal iterations of their old previous algorithms. At the end of this process, we will introduce the "Bubble Sort," "Insertion Sort," and "Heap Sort" algorithms and talk about the difference in their runtime. During the first hour, we will spend time rebuilding and adding onto programs from the day before. Then, we will deliver a lecture on state-of-the-art sorting techniques in the second hour.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #9:
Skills: Students will have a well-rounded and exceptional ability to solve extremely difficult problems. They will also have a few more important algorithms under their tool belt.
Goals: Students won't feel fear in the face of challenges or prompts they don't know how to answer. Instead, they will be able to execute actionable steps that will push them in the right direction of a solution. They will be able to leverage this technique in their everyday lives and become unstoppable creative problem solvers.

$\Delta\Sigma$ Day #10: Bringing it all Together & Taking Off

- **Problem Session #1 (10:00AM - 12:00PM):** On this final day, students will receive a wide variety of problems and mathematical puzzles ranging both in difficulty and topic. The topics will span the content of the entire program so students will be able to reference all of their techniques in action. During the first session, students will work in teams to solve the problems. Then, during the second hour, we will wrap up by going over the solutions and preparing for the final problem session later in the day.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

- **Problem Session #2 (4:00PM - 6:00PM):** students will receive a wide variety of coding challenges and ranging both in difficulty and topic. The topics will span the content of the entire program so students will be able to reference all of their techniques in action. During the first session, students will work in teams to solve the problems. Then, during the second hour, we will wrap up by going over the solutions and handing out awards and certificates.

- **Deliverables:** Lesson Overview handout with extra practice puzzles + 2-hour lesson recording for student reference. Both of these files are downloadable so students can reference their work beyond the program.

By the end of day #10:
Skills: Students will have many hours of problem solving experience, which will feed directly into their problem-solving and puzzling intuition skills.
Goals: Students will have a reformed, curious, and excited outlook on their education. They will have a set of techniques to be successful learners and creative problem solvers. They will be able to apply these skills in their everyday life and actively seek challenges to keep them academically stimulated

$\Delta\Sigma$ What we ask of students to succeed

After glancing over the curriculum, it may appear that Abstract Academy is a very ambitious summer program. However, our puzzles and problems are designed with the intent of inducing curiosity. Our problems spark are designed around "aha" moments which have shown to motivate learners.

After each lesson, students will receive a small set of homework problems ranging from three to five exercises. It is okay if they aren't able to solve all the problems. All that matters is that they reach out for help when stuck: they inquire on the Abstract Academy forums and bring their questions to class the next day.

Finally, please do not sign your child up for this program if they are not curious about puzzle-solving and creative thinking. This program is not meant to place bandaids on weaknesses, it's meant to empower and project creative thinkers. If they don't express an interest for puzzles, their summers are better spent doing what they love! ☺