Ohio's Learning Standards | Mathematics

Discrete Mathematics/Computer Science-DRAFT

Ohio | Department of Education

Table of Contents

Introduction To Ohio's Learning Standards For Mathematics.......3

Introduction To Ohio's Learning Standards For Computer Science (2022)....................5

A Note On Rigor And Algebra 2 Equivalency....................7

What Is Discrete Math?....................8

Standards For Mathematical Practice 9

Computer Science Practices[1]....................12

Equity And Computer Science[1]....................14

Mathematical Standards For High School....................17

Critical Areas Of Focus 20

Discrete Math/Computer Science Course Overview....................22

Mathematics 22

Computer Science 23

High School—Modeling....................24

High School—Algebra 26

Algebra Standards 27

High School—Functions 28

Functions Standards 29

High School—Geometry....................30

Geometry Standards....................32

High School—Statistics And Probability....................33

Statistics And Probability Standards....................34

Computer Science Grades 9-12—Foundational Level....................35

Computer Science Grades 9-12—Advanced Level 37

Glossary 38

Computer Science References....................40

Acknowledgments 41

Introduction to Ohio's Learning Standards for Mathematics

PROCESS

To better prepare students for college and careers, educators used public comments along with their professional expertise and experience to revise Ohio's Learning Standards. In spring 2016, the public gave feedback on the standards through an online survey. Advisory committee members, representing various Ohio education associations, reviewed all survey feedback and identified needed changes to the standards. Then they sent their directives to working groups of educators who proposed the actual revisions to the standards. The Ohio Department of Education sent their revisions back out for public comment in July 2016. Once again, the Advisory Committee reviewed the public comments and directed the Working Group to make further revisions. Upon finishing their work, the department presented the revisions to the Senate and House education committees as well as the State Board of Education.

Then in 2019, Ohio started the Strengthening Math Pathways Initiative. Two groups were formed: the Math Pathways Advisory Council and the Math Pathways Architects. The advisory council, made of representatives from education stakeholder groups, aligned systems and structures between secondary and postsecondary mathematics. The Math Pathways Architects, made up of high school and collegiate math faculty, aligned mathematics between the two systems. One of the outcomes was four proposed Algebra 2 equivalent courses: Quantitative Reasoning, Statistics and Probability, Data Science Foundations, and Discrete Math/Computer Science. A workgroup was formed for each of these courses. This document is the result of the Data Science Foundations Workgroup with oversight from the Math Pathways Architects.

UNDERSTANDING MATHEMATICS

These standards define what students should understand and be able to do in their study of mathematics. Asking a student to understand something means asking a teacher to assess whether the student has understood it. But what does mathematical understanding look like? One hallmark of mathematical understanding is the ability to justify, in a way appropriate to the student's mathematical maturity, why a particular mathematical statement is true, or where a mathematical rule comes from. There is a world of difference between a student who can summon a mnemonic device to expand a product such as $(a + b)(x + y)$ and a student who can explain where the mnemonic device comes from. The student who can explain the rule understands mathematics at a much deeper level. Then the student may have a better chance to succeed at a less familiar task such as expanding $(a + b + c)(x + y)$. Mathematical understanding and procedural skills are equally important, and both are assessable using mathematical tasks of sufficient richness.

The content standards are grade specific. However, they do not define the intervention methods or materials necessary to support students well below or above grade-level expectations. It is also beyond the scope of the standards to define the full range of supports appropriate for English learners and students with special needs. At the same time, all students must have the opportunity to learn and meet the same high standards if they are to access the knowledge and skills necessary in their post-school lives. Educators should read the standards allowing for the widest possible range of students to participate fully from the outset. They should provide appropriate accommodation to ensure maximum participation of students with special education needs. For example, schools should allow students with disabilities in reading to use Braille, screen reader technology or other assistive devices. Those with disabilities in writing should have scribes, computers or speech-to-text technology. In a similar vein, educators should interpret the speaking and listening standards broadly to include sign language.

No set of grade-specific standards can fully reflect the great variety in abilities, needs, learning rates and achievement levels of students in any given classroom. However, the standards do provide clear signposts along the way to help all students achieve the goal of college and career readiness.

The standards begin on page 9 with the eight Standards for Mathematical Practice.

Introduction to Ohio's Learning Standards for Computer Science (2022)

Substitute House Bill Number 170 took effect in March 2018, requiring the State Board of Education of Ohio to adopt standards and a model curriculum for grade K-12 instruction in computer science. A team of Ohio educators came together to develop and write the computer science standards and model curriculum, and the State Board adopted these in December 2018.

Ohio House Bill Number 110, passed in July 2021, included several new provisions for K-12 computer science education. The law requires the Ohio Department of Education to update the Ohio Learning Standards and Model Curriculum for Computer Science within one year of the effective date of HB 110 (Ohio Revised Code 3301.079(A)(4). Following the established process, a team of Ohio educators came together to develop and write the revisions for the computer science standards and the State Board adopted these in 2022.

Ohio's Standards in Computer Science are fully aligned to Ohio's five-year strategic plan for education, *Each Child, Our Future*. The strategic plan acknowledges a major education policy shift around technology. A student's ability to use technology strategically is now identified as foundational and just as important as mathematics and English language arts, from which all other learning is built.

GUIDING ASSUMPTIONS

The team of Ohio educators that developed the standards and model curriculum had a clear goal – to encourage districts and educators to give all Ohio students opportunities to learn computer science. Beginning in the earliest grades and continuing through grade 12, Ohio's students will develop a foundation of computer science knowledge and gain experiences in computational thinking and problem-solving to become creators and innovators of computing technology. Ohio's Computer Science Standards and Model Curriculum will give students experiences that help them discover and take part in a world continually influenced by technology and to understand the role of computing in that world.

OVERVIEW OF THE COMPUTER SCIENCE STANDARDS CONTENT

The standards will support a progression of learning in each core concept or strand to provide computer science experiences for all Ohio students. The K-8 standards integrate computer science into instruction across subject areas including mathematics, science, history, English language arts, fine arts, world language and career and technology courses. The high school computer science standards provide both foundational and advanced opportunities districts can use to design as separate courses or, when appropriate, integrate into other disciplines.

Ohio's Computer Science Standards and Model Curriculum are organized into the following strands:

Computing Systems—Addresses how devices, including hardware and software, interact to accomplish tasks and how students can troubleshoot computing systems when they do not work as intended.

Networks and the Internet—Addresses how devices and networks connect to share information and resources and how students can apply cybersecurity concepts to protect information.

Data and Analysis—Addresses how data can be collected and stored; analyzed and communicated; and used to make more accurate predictions.

Algorithmic Thinking and Programming—Addresses program development, including the use of algorithms, variables, control structures and modules.

Artificial Intelligence—Addresses machine learning, natural interaction, perception, representation and reasoning and societal impacts.

Impacts of Computing—Addresses computing's influence on our world by examining the relationship between computing and culture, computing's impact on social interaction, and legal and ethical implications of computing.

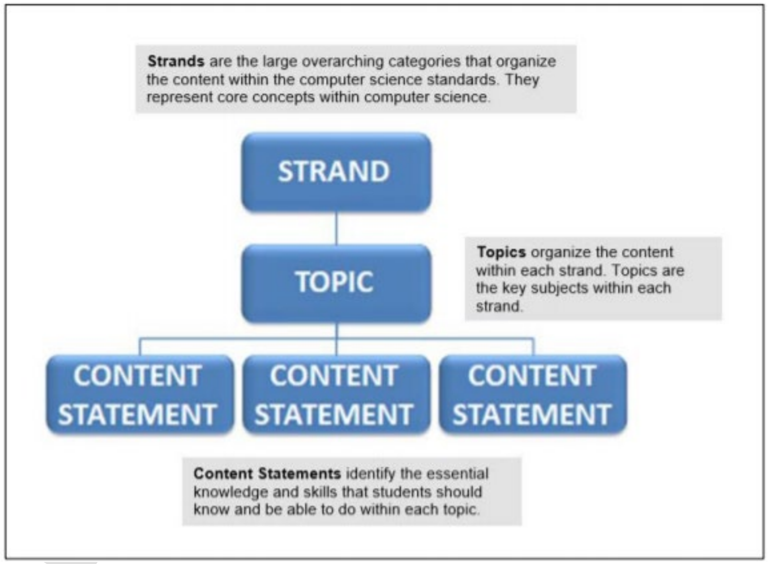
Computational Thinking is a problem-solving process that students use to engage with concepts in the computer science standards. This thinking involves formulating problems in a way that can be carried out by a computer. Using computational thinking to solve a problem includes breaking down the problem into manageable parts, recognizing patterns, excluding irrelevant details to abstract or identify general principles that generate these patterns and developing step-by-step sequences or algorithms to solve the problem and similar problems. Computational thinking can be applied with or without computers, for example, through "unplugged" activities. While computational thinking is a focus in computer science, it also is used in content areas beyond computer science.

It is important that computer science not be confused with other aspects and uses of computer technology in schools, including:

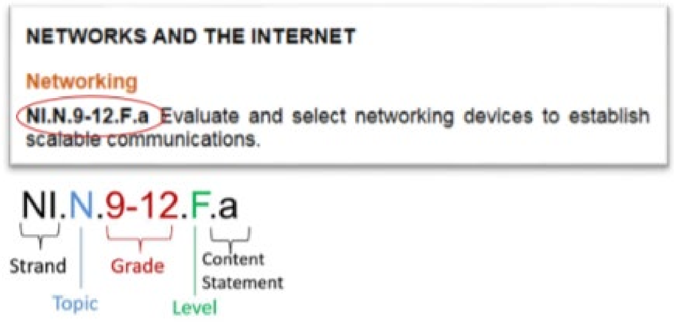
- **Computer literacy** "refers to the general use of computers and programs, such as productivity software." Examples of computer literacy include performing an internet search and creating a digital presentation.
- **Educational (computer) technology** "applies computer literacy to school subjects. For example, students in an English class can use a web-based application to collaboratively create, edit and store an essay online."
- **Digital citizenship** "refers to the appropriate and responsible use of technology, such as choosing an appropriate password and keeping it secure."
- **Information technology** "often overlaps with computer science but is mainly focused on industrial applications of computer science, such as installing [and operating] software rather than creating it. Information technology professionals often have a background in computer science."
(K-12 Computer Science Framework, 2016, pp.13-14)

OVERVIEW OF THE COMPUTER SCIENCE STANDARDS FRAMEWORK

Ohio's Computer Science Standards are organized by strands, topics and content statements.



Grades 9-12—Content statements are organized by grade band into two levels–Foundational and Advanced. See an example of a content statement for high school and its corresponding content statement code below. This content statement addresses the topic of Networking within the Networks and the Internet strand, at the Foundational Level.



A Note on Rigor and Algebra 2 Equivalency

Ohio law states that students must have four units of mathematics and that one of those units should be Algebra 2/Math 3 or its equivalent. Ohio has decided to expand guidance around what it means to be *equivalent* to Algebra 2.

It has been decided that *equivalent* refers to the level of rigor and reasoning, not content. There are many branches of mathematics that are equally rigorous but have different content focuses. All equivalent courses should have the same level of rigor and reasoning that are needed to be successful in an entry-level, credit-bearing postsecondary mathematics course.

Ohio has defined rigor as the following:

> "Students use mathematical language to communicate effectively and to describe their work with clarity and precision. Students demonstrate how, when, and why their procedure works and why it is appropriate. Students can answer the question, 'How do we know?'"

This can be illustrated in the table to the right.

Currently, four courses have been determined to be equivalent to Algebra 2: Advanced Quantitative Reasoning, Statistics and Probability, Data Science Foundations and Discrete Math/Computer Science. The same level of rigor applying to Algebra 2 equivalent courses should also apply to an Algebra 2/Mathematics 3 course. This document explains what should be included in an Algebra 2/Mathematics 3 course to prepare students for a Calculus-based STEM career.

Rigorous courses are…	vs	Rigorous courses are not…
Defined by complexity, which is a measure of the thinking, action, or knowledge that is needed to complete the task		Characterized by difficulty, which is a measure of effort required to complete a task
Measured in depth of understanding		Measured by the amount of work
Opportunities for precision in reasoning, language, definitions, and notation that are sufficient to appropriate age/course		Based on procedure alone
Determined by students' process		Measured by assigning difficult problems
Opportunities for students to make decisions in problem-solving		Defined only by the resources used
Opportunities to make connections		Taught in isolation
Supportive of the transfer of knowledge to new situations		Repetitive
Driven by students developing efficient explanations of solutions and why they work, providing opportunities for thinking and reasoning about contextual problems and situations		Focused on getting an answer
Defined by what the student does with what you give them		Defined by what you give the student

What is Discrete Math?

Discrete Mathematics is an area of mathematics that most closely connects with the field of computer science. It is the study of mathematical structures that are countable or otherwise distinct and separable (as opposed to continuous quantities like in algebra or calculus).

THE NEED FOR COMPUTER SCIENCE

The Computer Science Field is one of the fastest-growing and highest-paying career paths in Ohio. However, there is a limited supply of Ohio students interested in Computer Science. This is largely based on how exposed students are to computational thinking and computer science concepts. Additionally, educating students in computer science is beneficial for all students. With the digital age rising, there is a need to develop logical thinking and problem-solving which are all a part of learning computer science. To prepare students for this field, they also need exposure to advanced mathematics. Combining mathematics and computer science gives more students access to computer science concepts.

The purpose of a Discrete Math/Computer Science math course is to give students exposure to computer science in a way that connects with advanced mathematics instruction. The course is intended to spark student interest in careers involving computer science and technology. Since the calculus pathway is not meeting the needs of all students, a Discrete Math/Computer Science course provides another alternative that is rigorous and prepares students for a post-high school learning experience including an apprenticeship and/or a two-year or four-year college program or developing skills needed for a position in the field of computer programming.

EARNING SIMULTANEOUS CREDIT

A licensed high school mathematics or computer science teacher may teach this course. Students who take this course may earn credit in mathematics and computer science simultaneously. Follow the guidance in Ohio's Integrated Coursework for Simultaneous Credit.

Ohio's Strategic Plan, *Each Child, Our Future*, envisions that each child is challenged to discover and learn, prepared to pursue a fulfilling post-high school path and empowered to become a resilient, lifelong learner who contributes to society. Part of this empowerment can be found in the concept of integrated coursework. Integrating coursework allows students to have a unique meaningful learning experience. Strategy 10 of *Each Child, Our Future* emphasizes giving students multiple ways to demonstrate the knowledge, skills and dispositions for high school graduation and beyond. This includes redesigning middle and high schools to contribute to or create a more successful learning environment.

Ohio law allows districts, community schools and chartered nonpublic schools to integrate content standards from multiple subject areas into a single course for which students can earn simultaneous credit. Districts and schools can award *simultaneous credit* for multiple courses at once if the content from the courses is covered and mastered by a student through an integrated course. Integrating coursework *for simultaneous credit* allows students to have unique meaningful learning experiences that best meet their needs and interests.

The Integrating Coursework and Awarding Simultaneous Credit Guidance for Schools document outlines processes and considerations for schools when developing integrated courses. It also provides guidance for awarding appropriate credit for integrated coursework to satisfy Ohio's graduation requirements.

A yearlong course that focuses on integrating computer science standards with mathematics standards and concepts from discrete mathematics is eligible under integrated coursework for simultaneous credit. See the Integrating Coursework and Awarding Simultaneous Credit Guidance for Schools document for more information on the necessary criteria for a course to be eligible for simultaneous credit.

Standards for Mathematical Practice

The Standards for Mathematical Practice describe varieties of expertise that mathematics educators at all levels should seek to develop in their students. These practices rest on important "processes and proficiencies" with longstanding importance in mathematics education. The first of these are the NCTM process standards of problem-solving, reasoning and proof, communication, representation, and connections. The second is the strands of mathematical proficiency specified in the National Research Council's report *Adding It Up:* adaptive reasoning, strategic competence, conceptual understanding (comprehension of mathematical concepts, operations and relations), procedural fluency[G] (skill in carrying out procedures flexibly, accurately, efficiently, and appropriately), and productive disposition (habitual inclination to see mathematics as sensible, useful, and worthwhile, coupled with a belief in diligence and one's own efficacy).

1. Make sense of problems and persevere in solving them.
Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems and try special cases and simpler forms of the original problem to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondence between equations, verbal descriptions, tables and graphs or draw diagrams of important features and relationships, graph data and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solve more complicated problems and identify correspondences between different approaches.

2. Reason abstractly and quantitatively.
Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to *decontextualize*—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to *contextualize*, to pause as needed during the manipulation process to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them and knowing and flexibly using different properties of operations and objects.

3. Construct viable arguments and critique the reasoning of others.
Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They can analyze situations by breaking them into cases and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen to or read the

arguments of others, decide whether they make sense and ask useful questions to clarify or improve the arguments.

4. Model with mathematics.
Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community.

By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later.

They can identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts, and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5. Use appropriate tools strategically.
Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencils and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels can identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They can use technological tools to explore and deepen their understanding of concepts.

6. Attend to precision.
Mathematically proficient students try to communicate precisely with others. They try to use clear definitions in discussions with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently and express numerical answers with a degree of precision appropriate for the problem context. In elementary grades, students give carefully formulated explanations to each other. By the time they reach high school, they have learned to examine claims and make explicit use of definitions.

7. Look for and make use of structure.
Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7×8 equals the well-remembered $7 \times 5 + 7 \times 3$, in preparation for learning about the distributive property. In the expression $x^2 + 9x + 14$, older students can see the 14 as 2×7 and the 9 as $2 + 7$. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complex things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see $5 - 3(x - y)^2$ as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8. Look for and express regularity in repeated reasoning.
Mathematically proficient students notice if calculations are repeated and look both for general methods and shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, students might abstract the equation $^{(y-2)}/_{(x-1)} = 3$. Noticing the regularity in the way terms cancel when expanding $(x - 1)(x + 1)$, $(x - 1)(x^2 + x + 1)$, and $(x - 1)(x^3 + x^2 + x + 1)$ might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

CONNECTING THE STANDARDS FOR MATHEMATICAL PRACTICE TO THE STANDARDS FOR MATHEMATICAL CONTENT

The Standards for Mathematical Practice describe ways in which developing student practitioners of the discipline of mathematics increasingly ought to engage with the subject matter as they grow in mathematical maturity and expertise throughout the elementary, middle and high school years. Designers of curricula, assessments, and professional development should all attend to the need to connect mathematical practices to mathematical content in mathematics instruction.

The Standards for Mathematical Content are a balanced combination of procedure and understanding. Expectations that begin with the word “understand” are often especially good opportunities to connect the practices to the content. Students who lack understanding of a topic may rely on procedures too heavily. Without a flexible base from which to work, they may be less likely to consider analogous problems, represent problems coherently, justify conclusions, apply mathematics to practical situations, use technology mindfully to work with the mathematics, explain the mathematics accurately to other students, step back for an overview or deviate from a known procedure to find a shortcut. In short, a lack of understanding effectively prevents a student from engaging in mathematical practices. In this respect, those content standards which set an expectation of understanding are potential “points of intersection” between the Standards for Mathematical Content and the Standards for Mathematical Practice. These points of intersection are intended to be weighted toward central and generative concepts in the school mathematics curriculum that most merit the time, resources, innovative energies and focus necessary to qualitatively improve the curriculum, instruction, assessment, professional development and student achievement in mathematics.

Computer Science Practices[1]

Practice 1. Fostering an Inclusive Computing Culture
Building an inclusive and diverse computing culture requires strategies for incorporating perspectives from people of different genders, ethnicities and abilities. Incorporating these perspectives involves understanding the personal, ethical, social, economic and cultural contexts in which people operate. Considering the needs of diverse users during the design process is essential to producing inclusive computational products.

By the end of Grade 12, students should be able to do the following:

- Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.
- Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.
- Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.

Practice 2: Collaborating Around Computing
Collaborative computing is the process of performing a computational task by working in pairs and teams. Because it involves asking for the contributions and feedback of others, effective collaboration can lead to better outcomes than working independently. Collaboration requires individuals to navigate and incorporate diverse perspectives, conflicting ideas, disparate skills and distinct personalities. Students should use collaborative tools to effectively work together and to create complex artifacts. By the end of Grade 12, students should be able to do the following:

- Cultivate working relationships with individuals possessing diverse perspectives, skills and personalities.
- Create team norms, expectations and equitable workloads to increase efficiency and effectiveness.
- Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.
- Evaluate and select technological tools that can be used to collaborate on a project.

Practice 3: Recognizing and Defining Computational Problems
The ability to recognize appropriate and worthwhile opportunities to apply computation is a skill that develops over time and is central to computing. Solving a problem with a computational approach requires defining the problem, breaking it down into parts and evaluating each part to determine whether a computational solution is appropriate.

By the end of Grade 12, students should be able to do the following:

- Identify complex, interdisciplinary, real-world problems that can be solved computationally.
- Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. Compose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.
- Evaluate whether it is appropriate and feasible to solve a problem computationally.

[1] The "Computer Science Practices" section has been modified from chapter two of the K-12 Computer Science Framework Statements by Grade Band, (K–12 Computer Science Framework. (2016). Retrieved from https://k12cs.org/framework-statements-by-grade-band/This work is licensed under Creative Commons (CC BY-NC-SA 4.0).

Practice 4: Developing and Using Abstractions
Abstractions are formed by identifying patterns and extracting common features from specific examples to create generalizations. Using generalized solutions and parts of solutions designed for broad reuse simplifies the development process by managing complexity.

By the end of Grade 12, students should be able to do the following:

- Extract common features from a set of interrelated processes or complex phenomena.
- Evaluate existing technological functionalities and incorporate them into new designs.
- Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.
- Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.

Practice 5: Creating Computational Artifacts
The process of developing computational artifacts embraces both creative expression and the exploration of ideas to create prototypes and solve computational problems. Students create artifacts that are personally relevant or beneficial to their community and beyond. Computational artifacts can be created by combining and modifying existing artifacts or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems and apps.

By the end of Grade 12, students should be able to do the following:

- Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints and user expectations.
- Create a computational artifact for practical intent, personal expression, or to address a societal issue.
- Modify an existing artifact to improve or customize it.

Practice 6: Testing and Refining Computational Artifacts
Testing and refinement are the deliberate and iterative process of improving a computational artifact. This process includes debugging (identifying and fixing errors) and comparing actual outcomes to intended outcomes. Students also respond to the changing needs and expectations of end users and improve the performance, reliability, usability, and accessibility of artifacts.

By the end of Grade 12, students should be able to do the following:

- Systematically test computational artifacts by considering all scenarios and using test cases.
- Identify and fix errors using a systematic process.
- Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability and accessibility.

Practice 7: Communicating About Computing
Communication involves personal expression and exchanging ideas with others. In computer science, students communicate with diverse audiences about the use and effects of computation and the appropriateness of computational choices. Students write clear comments, document their work, and communicate their ideas through multiple forms of media. Clear communication includes using precise language and carefully considering possible audiences.

By the end of Grade 12, students should be able to do the following:

- Select, organize and interpret large data sets from multiple sources to support a claim.
- Describe, justify and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.
- Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.

EQUITY AND COMPUTER SCIENCE[1]

COMPUTER SCIENCE FOR ALL

To help realize the vision of computer science for all students, equity must be at the forefront of the state's efforts to implement computer science standards. Equity is more than whether classes are available. It includes how those classes are taught, how students are recruited and how the classroom culture supports diverse learners and promotes retention. When equity exists, schools expect academic success for every student and make that success accessible to every student. The result of such equity is a classroom of diverse students based on factors such as race, gender, disability, socioeconomic status and English language proficiency. All these students have high expectations and feel empowered to learn.

Computer science faces intense challenges related to access, opportunity and culture.

- The 2021 State of Computer Science report showed that only 51 percent of public high schools offer foundational computer science courses (Code.org et al, 2021). This data showed that students with the least access are Black, Latino and Native American, from lower-income backgrounds and urban and rural areas.

Even when computer science courses are available, there are wide gaps in participation and the level of instruction.

- For the 2020 Advanced Placement (AP) Computer Science exam, only 31 percent of students were female, 6 percent were Black or African American, 16 percent were Hispanic or Latino and 0.5 percent were Native American (College Board, 2020).
- The potential impact of these gaps in participation is illustrated in the statistic that females who take high school AP Computer Science are 10 times more likely to major in computer science in college than students who do not take this course (Morgan & Klaric, 2007).
- Especially in schools with large numbers of African American and Latino students, computer classes too commonly offer only basic, rudimentary user skills rather than engaging students in the problem-solving and computational thinking practices that form the foundation of computer science (Margolis et al., 2012).

[1] The "Equity and Computer Science" section has been modified from chapter two of the K-12 Computer Science Framework, "Equity in Computer Science Education." (K–12 Computer Science Framework. (2016).

The lack of representation in computer science after K–12 reflects the lack of access and participation in grades K–12. In 2021, only 26.2 percent of workers employed in computer and mathematical occupations were female. Only 8.5 percent were Black or African American, and only 8.3 percent were Hispanic or Latino (Bureau of Labor Statistics, 2021).

EFFORTS TO INCREASE EQUITY

Even when schools have made computer science courses available to students, inequity can be perpetuated at the classroom level. Educators can work to ensure equity through changes in curriculum, instruction and classroom culture.

- Educators can reach *students with disabilities* using learning accommodations, curricular modifications and established techniques for differentiated instruction. For example, the Quorum programming language accommodates students with visual impairments by enabling the programming language to be read by computer screen readers (Quorum, 2019). Recent research shows ways to use Universal Design for Learning (UDL) to develop and refine introductory computer science experiences for a wider range of learners (Hansen et al., 2016). Educators also can apply instructional strategies used in other content areas to support struggling learners and students with disabilities. For example, if verbal prompting helps in math instruction, it will likely help in computer science instruction (Snodgrass, Israel, & Reese, 2016).

- A variety of approaches make programming more accessible *to young learners and beginners.* Visual, block-based programming languages allow students to program without the obstacle of syntax errors (errors in typing commands) found in traditional text-based languages. Programming environments on tablets have made programming even more accessible to younger children by reducing the number of available commands and the amount of reading required to navigate the options (Strawhacker & Bers, 2014).
- To address a *lack of computer and internet access*, educators can help students learn many computer science topics, such as algorithmic thinking, searching, sorting and logic through "unplugged" activities. Ohio's initiative, InnovateOhio, strives to transform Ohio's communities and bring opportunity for growth in the rural and urban areas through a statewide broadband strategy to improve access to high-speed internet (InnovateOhio, 2022).
- To reach *females and underrepresented minorities*, teachers can use strategies to work against issues such as the threat of stereotyping or bias. For example, stereotype threats can be mitigated by altering the wording of test questions to be gender-neutral and using examples that are equally relevant to both males and females (Kumar, 2012). It also is important for students to have diverse role models in the field so they can imagine themselves as a computer scientist. Role models also help dispel stereotypes of how computer scientists look and act (Goode, 2008).

Below, are other practices that teachers can adopt and adapt to change classroom culture and broaden participation in computer science:

- Connect computer science to concepts that motivate children, like fairness and social justice (Denner et al., 2015).
- Practice culturally relevant pedagogy to tie computer science to students' experiences, culture and interests (Margolis et al., 2014). Designing projects and instruction to be socially relevant and meaningful for diverse students helps them "build personal relationships with CS concepts and applications -- an important process for discovering the relevance of CS for their own lives." (Margolis et. al, 2012, p. 76)
- Reflect on beliefs and actions to address stereotypes among students and teachers (Margolis et al., 2014).

EQUITY AND THE COMPUTER SCIENCE STANDARDS

The computer science standards reflect the writing team's considerations on equity. The standards describe concepts and skills all students can benefit from, regardless of whether they go on to postsecondary computer-science education or a career in computer science.

Equity is woven into the computer science concepts and skills across grade levels. This is especially apparent in the core concept or strand involving *Impacts of Computing*. Here, students examine the social implications of the digital world, including their impact on equity and access to computing. Specific content statements address equity directly. For example, in grade 3, students identify diverse user needs and "how computing devices have built-in features to increase accessibility to all users." In grade 7, students "evaluate various technologies to identify issues of bias and accessibility." Students in grade 8 build on prior learning to work against existing inequities, they propose guidelines "to positively impact bias and accessibility in the design of future technologies."

As students design computational products, they engage in computer science practices that also directly involve consideration of equity, inclusion and diversity. Students foster inclusion as they develop products that “include the unique perspectives of others” and “address the needs of diverse end users.” Students encourage diversity through working in teams “with individuals possessing diverse perspectives.” Involving students in such practices stresses the need to practice equity when doing computer science. Through such practices, students can see the benefit of, for example, considering the products they develop from the perspectives of a diverse group of end-users, such as those with visual impairments and English language learners.

Mathematical Standards for High School

PROCESS

The high school standards specify the mathematics that all students should study to be college and career ready. Additional mathematics that students should learn to take advanced courses such as calculus, advanced statistics or discrete mathematics is indicated by (+), as in this example:

> *(+) Represent complex numbers on the complex plane in rectangular and polar form (including real and imaginary numbers).*

All standards without a (+) symbol should be in the common mathematics curriculum for all college and career-ready students. Standards with a (+) symbol may also appear in courses intended for all students. However, standards with a (+) symbol will not appear on Ohio's State Tests.

- The high school standards are listed in conceptual categories:
- Modeling
- Number and Quantity
- Algebra
- Functions
- Geometry
- Statistics and Probability

Conceptual categories portray a coherent view of high school mathematics; a student's work with functions, for example, crosses several traditional course boundaries, potentially up through and including Calculus.

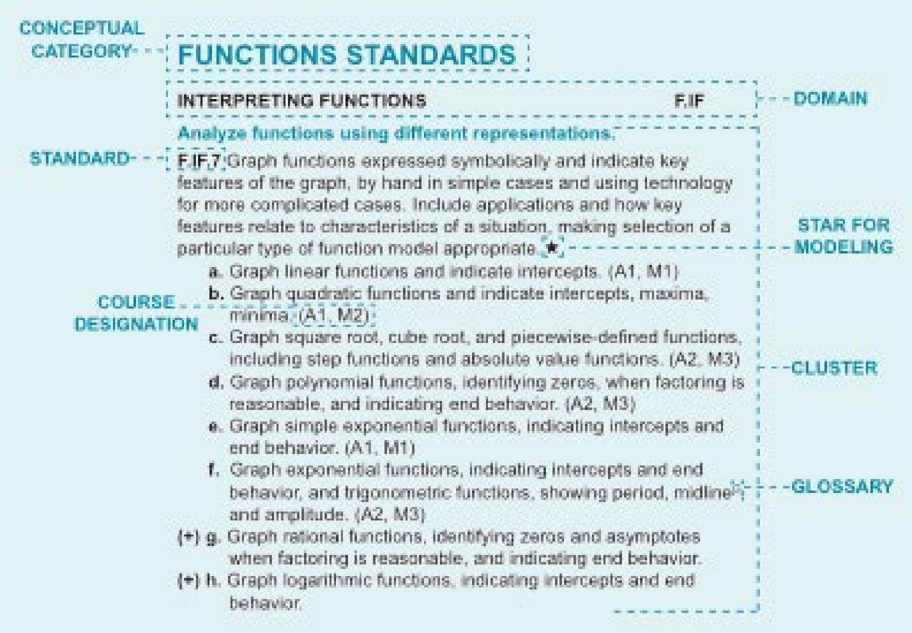
Modeling is best interpreted not as a collection of isolated topics but in relation to other standards. Making mathematical models is a Standard for Mathematical Practice and specific modeling standards appear throughout the high school standards indicated by a star symbol (★).

Proofs in high school mathematics should not be limited to geometry. Mathematically proficient high school students employ multiple proof methods, including algebraic derivations, proofs using coordinates, and proofs based on geometric transformations, including symmetries. These proofs are supported by the use of diagrams and dynamic software and are written in multiple formats including not just two-column proofs but also proofs in paragraph form, including mathematical symbols. In statistics, rather than using mathematical proofs, arguments are made based on empirical evidence within a properly designed statistical investigation.

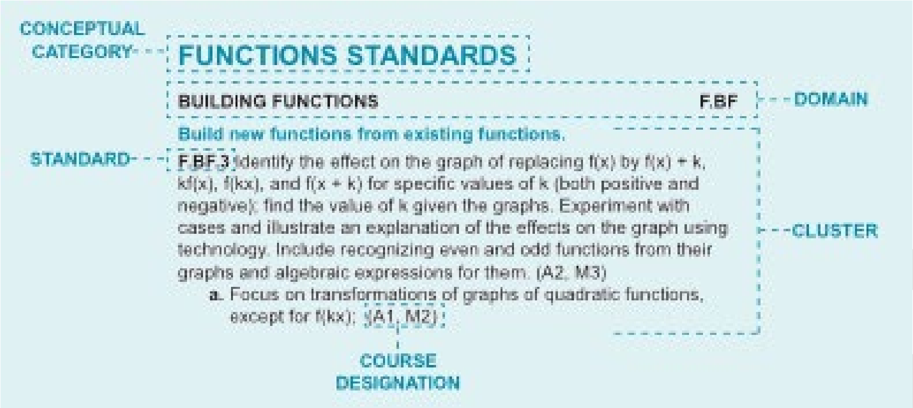
HOW TO READ THE HIGH SCHOOL CONTENT STANDARDS

- **Conceptual Categories** are areas of mathematics that cross through various course boundaries.
- **Standards** define what students should understand and be able to do.
- **Clusters** are groups of related standards. Note that standards from different clusters may sometimes be closely related because mathematics is a connected subject.
- **Domains** are larger groups of related standards. Standards from different domains may sometimes be closely related.
- [G] shows there is a definition in the glossary for this term.
- (★) indicates that modeling should be incorporated into the standard. (See the Conceptual Category of Modeling pages 12-13)
- (+) indicates that it is a standard for students who are planning on taking advanced courses. Standards with a (+) sign will not appear on Ohio's State Tests.

Some standards have course designations such as (A1, M1) or (A2, M3) listed after an **a., b.** or **c.** These designations help teachers know where to focus their instruction within the standard. In the example below the beginning section of the standard is the stem. The stem shows what the teacher should be doing for all courses. (Notice in the example below that modeling (★) should also be incorporated.) Looking at the course designations, an Algebra 1 teacher should be focusing his or her instruction on **a.** which focuses on linear functions; **b.** which focuses on quadratic functions; and **e.** which focuses on simple exponential functions. An Algebra 1 teacher can ignore **c., d.** and **f,** as the focuses of these types of functions will come in later courses. However, a teacher may choose to touch on these types of functions to extend a topic if he or she wishes.



Notice that in the standard below, the stem has a course designation. This shows that the full extent of the stem is intended for an Algebra 2 or Math 3 course. However, **a.** shows that Algebra 1 and Math 2 students are responsible for a modified version of the stem that focuses on transformations of quadratics functions and excludes the *f(kx)* transformation. However, again a teacher may choose to touch on different types of functions besides quadratics to extend a topic if he or she wishes.



Critical Areas of Focus

CRITICAL AREA OF FOCUS #1

Communication and Analysis

Within this critical area students develop conclusions based on quantitative information and critical thinking. They recognize, make, and evaluate underlying assumptions in estimation, modeling, and data analysis. Students then organize and present thoughts and processes using mathematical evidence. They communicate clear and complete information in such a way that the reader or listener can understand the contextual and quantitative information in a situation. Students demonstrate numerical reasoning orally and in writing coherent statements and paragraphs.

In the context of real-world applications, students make and investigate mathematical conjectures. They can defend their conjectures and respectfully question conjectures made by their classmates. This leads to the development of mathematical arguments and informal proofs, which are ways of expressing particular kinds of reasoning and justification. Explanations (oral and written) include mathematical arguments and rationales, not just procedural descriptions or summaries. Listening to others' explanations gives students opportunities to develop their own understandings. Through communication, ideas become objects of reflection, refinement, discussion, and amendment. When students are challenged to communicate the results of their thinking to others orally or in writing, they learn to be clear, convincing and precise in their use of mathematical language. Additionally, conversations in which mathematical ideas are explored from multiple perspectives help the participants sharpen their thinking and make connections. **This critical area of focus cross-cuts all the other critical areas of focus.**

CRITICAL AREA OF FOCUS #2

Combinatorial Games

This unit introduces and explores functions, algorithmic thinking, inference, modeling and computer programming concepts through a variety of combinatorial games. In a combinatorial game:

- Two or more players alternate moves.
- There must be no chance involved in the game, and both players must have complete information about all aspects of the game at all times.
- On each move, the player whose turn it is, must have a finite number of possible actions.
- After a finite number of turns, each game must have a designated "end" where one player is the "winner".

Student discourse around a "winning strategy" is a central component of this unit as they analyze each game. Constructing viable arguments, demonstrating repeated reasoning, modeling with mathematics and critiquing the reasoning of others are critical components of the unit as students develop an understanding of *why* a strategy works and determine whether the strategy will always work.

CRITICAL AREA OF FOCUS #3

Counting and Combinatorics

Combinatorics is a branch of mathematics dealing with combinations of objects belonging to a finite set with certain constraints. In this unit, *combinatorics* are used to explore new real-world applications and extend understanding of previous work such as recursive formulas, exponential functions and sets. Students develop an understanding of the importance of labeling objects in a set with unique names. Models such as trees, lists, diagrams, tables and formulas are used to explore the concept of overcounting and its applications. Students refine their methods to be more efficient as they keep track of the number of objects in a set. Students construct meaningful mathematical justifications for a variety of counting formulas.

CRITICAL AREA OF FOCUS #4

Probability

Content associated with Statistics and Probability has a wide range of applications in many degree programs and careers. To prepare students for these careers, this unit is designed to highlight discrete mathematics and computer science connections to probability.

The probability tasks in the unit develop students’ reasoning associated with the probability of a finite number of events. Students use JavaScript® to apply discrete math and computer science concepts to predict and verify prescribed outcomes through increasing amounts of simulated trials. Students construct mathematical arguments while looking for patterns in different mathematical models and simulations to determine whether the model or simulated results are consistent with the outcome(s) of the event.

CRITICAL AREA OF FOCUS #5

Connectivity and Graph Theory

A graph is a connected graph if, for each pair of vertices, there exists at least one single path which joins them.

Vertex-edge graphs and their properties are introduced and explored through a variety of real-world applications. These types of graphs can be represented by a table, drawing or list where vertices (nodes or “dots”) may represent the objects, people or places while relationships among them are represented by edges (connecting lines or curves). Applications of contexts allow students to explore foundational concepts related to both fully and minimally connected graphs along with their properties. Students extend their understanding of the ways (and how) types of graphs can be useful tools for tracking relationships and specific connections among objects or people. Mathematical structures such as weighted planar, directed and undirected graphs for modeling pairwise relations are introduced as part of this exploration and application.

CRITICAL AREA OF FOCUS #6

Iteration and Recursion

Recursion is when an object is defined in terms of itself or objects of the same type. While recursive functions repeat by calling on themselves, iteration functions repeat using a set of directions. In this unit, students work flexibly with iterations and recursion as they generalize patterns to identify efficient processes in a variety of applications. Previous understanding of functions is extended to include recursive and explicit applications through real-world contexts. Factorial functions are introduced and explored through work with iteration and recursive functions.

CRITICAL AREA OF FOCUS #7

Cryptography

Cryptography is the human endeavor of trying to maintain privacy in communication. Students use inverse functions, combinations and permutations to encrypt and decrypt messages. The concept of *modulus* or *mods* is introduced as students explore remainders through a variety of computer science applications. Students extend their understanding of non-Base-10 numbers through work with binary, 5-digit and 6-digit number systems to discover their limitations and applications in cryptography.

Discrete Math/Computer Science Course Overview

MATHEMATICS

ALGEBRA

SEEING STRUCTURE IN EXPRESSIONS

- Interpret the structure of expressions.
- Write expressions in equivalent forms to solve problems.

CREATING EQUATIONS

- Create equations that describe numbers or relationships.

FUNCTIONS

INTERPRETING FUNCTIONS

- Understand the concept of a function and use function notation.
- Interpret functions that arise in applications in terms of the context.

BUILDING FUNCTIONS

- Build a function that models a relationship between two quantities.

LINEAR, QUADRATIC, AND EXPONENTIAL MODELS

- Construct and compare linear, quadratic and exponential models and solve problems.
- Interpret expressions for functions in terms of the situation they model.

GEOMETRY

CONGRUENCE

- Experiment with transformations in the plane.
- Understand congruence in terms of rigid motions.
- Prove geometric theorems both formally and informally using a variety of methods.
- Make geometric constructions.
- Classify and analyze geometric figures.

MATHEMATICAL PRACTICES

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

MODELING IN GEOMETRY

- Apply geometric concepts in modeling situations.

STATISTICS AND PROBABILITY

INTERPRETING CATEGORICAL AND QUANTITATIVE DATA

- Summarize, represent and interpret data on a single count or measurement variable.
- Summarize, represent and interpret data on two categorical and quantitative variables.
- Interpret linear models.

MAKING INFERENCES AND JUSTIFYING CONCLUSIONS

- Understand and evaluate random processes underlying statistical experiments.
- Make inferences and justify conclusions from sample surveys, experiments and observational studies.

CONDITIONAL PROBABILITY AND THE RULES OF PROBABILITY

- Understand independence and conditional probability and use them to interpret data.

USING PROBABILITY TO MAKE DECISIONS

- Use probability to evaluate outcomes of decisions

COMPUTER SCIENCE

COMPUTER SCIENCE GRADES 9-12 FOUNDATIONAL LEVEL

NETWORKS AND THE INTERNET

- Networking
- Cybersecurity

DATA AND ANALYSIS

- Data Collection and Storage
- Visualization and Communication
- Inference and Modeling

ALGORITHMIC THINKING AND PROGRAMMING

- Algorithms
- Control Structures
- Variables and Data Representation
- Modularity

ARTIFICIAL INTELLIGENCE

- Representation & Reasoning

IMPACTS OF COMPUTING

- Social Interaction
- Safety, Law and Ethics

Computer Science Practices

1. **Fostering an Inclusive Computing Culture**
2. **Collaborating Around Computing**
3. **Recognizing and Defining Computational Problems**
4. **Developing and Using Abstractions**
5. **Creating Computational Artifacts**
6. **Testing and refining Computational Artifacts**
7. **Communicating About Computing**

COMPUTER SCIENCE GRADES 9-12 ADVANCED LEVEL

NETWORKS AND THE INTERNET

- Cybersecurity

DATA AND ANALYSIS

- Visualization and Communication

ALGORITHMIC THINKING AND PROGRAMMING

- Algorithms
- Control Structures
- Variables and Data Representation
- Modularity

ARTIFICIAL INTELLIGENCE

- Representation & Reasoning

IMPACTS OF COMPUTING

- Safety, Law and Ethics

High School—Modeling

Modeling links classroom mathematics and statistics to everyday life, work, and decision-making. Modeling is the process of choosing and using appropriate mathematics and statistics to analyze empirical situations, to understand them better and to improve decisions. Quantities and their relationships in physical, economic, public policy, social and everyday situations can be modeled using mathematical and statistical methods. When making mathematical models, technology is valuable for varying assumptions, exploring consequences and comparing predictions with data.

A model can be very simple, such as writing total cost as a product of unit price and number bought or using a geometric shape to describe a physical object like a coin. Even such simple models involve making choices. It is up to us whether to model a coin as a three-dimensional cylinder or whether a two-dimensional disk works well enough for our purposes. Other situations—modeling a delivery route, a production schedule or a comparison of loan amortizations—need more elaborate models that use other tools from the mathematical sciences. Real-world situations are not organized and labeled for analysis; formulating tractable models, representing such models, and analyzing them is appropriately a creative process. Like every such process, this depends on acquired expertise as well as creativity.

Some examples of such situations might include the following:

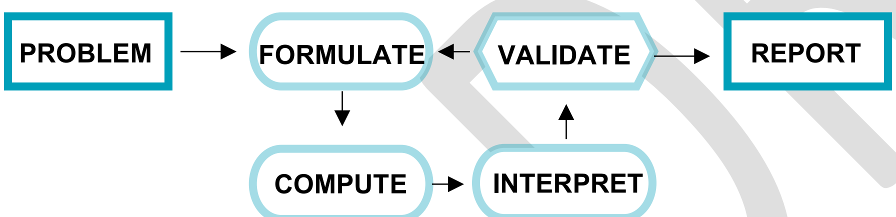
- Estimating how much water and food is needed for emergency relief in a devastated city of 3 million people, and how it might be distributed.
- Planning a table tennis tournament for 7 players at a club with 4 tables, where each player plays against each other.
- Designing the layout of the stalls at a school fair to raise as much money as possible.
- Analyzing the stopping distance for a car.
- Modeling savings account balance, bacterial colony growth or investment growth.
- Engaging in critical path analysis, e.g., applied to the turnaround of an aircraft at an airport.
- Analyzing risk in situations such as extreme sports, pandemics and terrorism.
- Relating population statistics to individual predictions.

In situations like these, the models devised depend on several factors: How precise an answer do we want or need? What aspects of the situation do we most need to understand, control or optimize? What resources of time and tools do we have? The range of models that we can create and analyze is also constrained by the limitations of our mathematical, statistical and technical skills, and our ability to recognize significant variables and relationships among them. Diagrams of various kinds, spreadsheets and other technology and algebra are powerful tools for understanding and solving problems drawn from different types of real-world situations.

One of the insights provided by mathematical modeling is that essentially the same mathematical or statistical structure can sometimes model seemingly different situations. Models can also shed light on the mathematical structures themselves, for example, as when a model of bacterial growth makes more vivid the explosive growth of the exponential function.

The basic modeling cycle is summarized in the diagram. It involves (1) identifying variables in the situation and selecting those that represent essential features, (2) formulating a model by creating and selecting geometric, graphical, tabular, algebraic or statistical representations that describe relationships between the variables, (3) analyzing and performing operations on these relationships to draw conclusions, (4) interpreting the results of the mathematics in terms of the original situation, (5) validating the conclusions by comparing them with the situation, and then either improving the model or, if it is acceptable, (6) reporting on the conclusions and the reasoning behind them. Choices, assumptions and approximations are present throughout this cycle.

In descriptive modeling, a model simply describes the phenomena or summarizes them in a compact form. Graphs of observations are a familiar descriptive model—for example, graphs of global temperature and atmospheric CO_2 over time.



Analytic modeling seeks to explain data based on deeper theoretical ideas, albeit with parameters that are empirically based; for example, exponential growth of bacterial colonies (until cut-off mechanisms such as pollution or starvation intervene) follows from a constant reproduction rate. Functions are an important tool for analyzing such problems.

Graphing utilities, spreadsheets, computer algebra systems and dynamic geometry software are powerful tools that can be used to model purely mathematical phenomena, e.g., the behavior of polynomials as well as physical phenomena.

MODELING STANDARDS

Modeling is best interpreted not as a collection of isolated topics but rather in relation to other standards. Making mathematical models is a Standard for Mathematical Practice, and specific modeling standards appear throughout the high school standards indicated by a star symbol (★).

High School—Algebra

EXPRESSIONS

An expression is a record of a computation with numbers, symbols that represent numbers, arithmetic operations, exponentiation and at more advanced levels, the operation of evaluating a function. Conventions about the use of parentheses and the order of operations assure that each expression is unambiguous. Creating an expression that describes a computation involving a general quantity requires the ability to express the computation in general terms, abstracting from specific instances.

Reading an expression with comprehension involves analysis of its underlying structure. This may suggest a different but equivalent way of writing the expression that exhibits some different aspects of its meaning. For example, $p + 0.05p$ can be interpreted as the addition of a 5% tax to a price p. Rewriting $p + 0.05p$ as $1.05p$ shows that adding a tax is the same as multiplying the price by a constant factor.

Algebraic manipulations are governed by the properties of operations and exponents, and the conventions of algebraic notation. At times, an expression is the result of applying operations to simpler expressions. For example, $p + 0.05p$ is the sum of the simpler expressions p and $0.05p$. Viewing an expression as the result of an operation on simpler expressions can sometimes clarify its underlying structure.

A spreadsheet or a computer algebra system (CAS) can be used to experiment with algebraic expressions, perform complicated algebraic manipulations and understand how algebraic manipulations behave.

EQUATIONS AND INEQUALITIES

An equation is a statement of equality between two expressions, often viewed as a question asking for which values of the variables the expressions on either side are equal. These values are the solutions to the equation. An identity, in contrast, is true for all values of the variables; identities are often developed by rewriting an expression in an equivalent form.

The solutions of an equation in one variable form a set of numbers; the solutions of an equation in two variables form a set of ordered pairs of numbers, which can be plotted in the coordinate plane. Two or more equations and/or inequalities form a system. A solution for such a system must satisfy every equation and inequality in the system.

An equation can often be solved by successively deducing from it one or more simpler equations. For example, one can add the same constant to both sides without changing the solutions but squaring both sides might lead to extraneous solutions. Strategic competence in solving includes looking ahead for productive manipulations and anticipating the nature and number of solutions.

Some equations have no solutions in a given number system but have a solution in a larger system. For example, the solution of $x + 1 = 0$ is an integer, not a whole number; the solution of $2x + 1 = 0$ is a rational number, not an integer; the solutions of $x^2 - 2 = 0$ are real numbers, not rational numbers; and the solutions of $x^2 + 2 = 0$ are complex numbers, not real numbers.

The same solution techniques used to solve equations can be used to rearrange formulas. For example, the formula for the area of a trapezoid, $A = (\frac{(b_1+b_2)}{2})h$, can be solved for h using the same deductive process.

Inequalities can be solved by reasoning about the properties of inequality. Many of the properties of equality continue to hold for inequalities and can be useful in solving them.

CONNECTIONS WITH FUNCTIONS AND MODELING

Expressions can define functions, and equivalent expressions define the same function. Asking when two functions have the same value for the same input leads to an equation; graphing the two functions allows for finding approximate solutions to the equation. Converting a verbal description to an equation, inequality or system of these is an essential skill in modeling.

Algebra Standards

SEEING STRUCTURE IN EXPRESSIONS A.SSE

Interpret the structure of expressions.

A.SSE.1. Interpret expressions that represent a quantity in terms of its context. ★

- **a.** Interpret parts of an expression, such as terms, factors, and coefficients.
- **b.** Interpret complicated expressions by viewing one or more of their parts as a single entity.

Write expressions in equivalent forms to solve problems.

A.SSE.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression. ★

- **c.** Use the properties of exponents to transform expressions for exponential functions. *For example, 8^t can be written as 2^{3t}.*

CREATING EQUATIONS A.CED

Create equations that describe numbers or relationships.

A.CED.1 Create equations and inequalities in one variable and use them to solve problems. Include equations and inequalities arising from linear, quadratic, simple rational, and exponential functions. ★

- **c.** Extend to include more complicated function situations with the option to solve with technology. (A2, M3)

High School—Functions

Functions describe situations where one quantity determines another. For example, the return on $10,000 invested at an annualized percentage rate of 4.25% is a function of the length of time the money is invested. Because we continually make theories about dependencies between quantities in nature and society, functions are important tools in the construction of mathematical models.

In school mathematics, functions usually have numerical inputs and outputs and are often defined by an algebraic expression. For example, the time in hours it takes for a car to drive 100 miles is a function of the car's speed in miles per hour, v; the rule $T(v) = {}^{100}/_{v}$ expresses this relationship algebraically and defines a function whose name is T.

The set of inputs to a function is called its domain. We often infer the domain to be all inputs for which the expression defining a function has a value, or for which the function makes sense in a given context.

A function can be described in various ways, such as by a graph, e.g., the trace of a seismograph; by a verbal rule, as in, "I'll give you a state, you give me the capital city;" by an algebraic expression like $f(x) = a + bx$; or by a recursive rule. The graph of a function is often a useful way of visualizing the relationship of the function models and manipulating a mathematical expression for a function can throw light on the function's properties.

Functions presented as expressions can model many important phenomena. Two important families of functions characterized by laws of growth are linear functions, which grow at a constant rate, and exponential functions, which grow at a constant percent rate. Linear functions with a constant term of zero describe proportional relationships.

A graphing utility or a computer algebra system can be used to experiment with the properties of these functions and their graphs and to build computational models of functions, including recursively defined functions.

CONNECTIONS TO EXPRESSIONS, EQUATIONS, MODELING AND COORDINATES.
Determining an output value for a particular input involves evaluating an expression; finding inputs that yield a given output involves solving an equation. Questions about when two functions have the same value for the same input lead to equations, whose solutions can be visualized from the intersection of their graphs. Because functions describe relationships between quantities, they are frequently used in modeling. Sometimes functions are defined by a recursive process, which can be displayed effectively using a spreadsheet or other technology.

Functions Standards

INTERPRETING FUNCTIONS F.IF

Understand the concept of a function and use function notation.

F.IF.2 Use function notation, evaluate functions for inputs in their domains, and interpret statements that use function notation in terms of a context.

F.IF.3 Recognize that sequences are functions, sometimes defined recursively, whose domain is a subset of the integers. *For example, the Fibonacci sequence is defined recursively by $f(0) = f(1) = 1$,* $f(n + 1) = f(n) + f(n - 1)$ for $n \geq 1$.

BUILDING FUNCTIONS F.BF

Build a function that models a relationship between two quantities.

F.BF.1 Write a function that describes a relationship between two quantities.★

- **b.** Combine standard function types using arithmetic operations. *For example, build a function that models the temperature of a cooling body by adding a constant function to a decaying exponential, and relate these functions to the model.* (A2, M3)
- **c.** Compose functions. *For example, if T(y) is the temperature in the atmosphere as a function of height, and h(t) is the height of a weather balloon as a function of time, then T(h(t)) is the temperature at the location of the weather balloon as a function of time.*

F.BF.2 Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.★

Build new functions from existing functions.

F.BF.4 Find inverse functions.

- **b.** Read values of an inverse function from a graph or a table, given that the function has an inverse. (A2, M3)
- **c.** Verify by composition that one function is the inverse of another. (A2, M3)
- **d.** Find the inverse of a function algebraically, given that the function has an inverse. (A2, M3)

LINEAR, QUADRATIC, AND EXPONENTIAL MODELS F.LE

Construct and compare linear, quadratic and exponential models, and solve problems.

F.LE.1 Distinguish between situations that can be modeled with linear functions and with exponential functions.★

- **a.** Show that linear functions grow by equal differences over equal intervals and that exponential function grow by equal factors over equal intervals.
- **b.** Recognize situations in which one quantity changes at a constant rate per unit interval relative to another.
- **c.** Recognize situations in which a quantity grows or decays by a constant percent rate per unit interval relative to another.

Interpret expressions for functions in terms of the situation they model.

F.LE.5 Interpret the parameters in a linear or exponential function in terms of a context.★

High School—Geometry

An understanding of the attributes and relationships of geometric objects can be applied in diverse contexts—interpreting a schematic drawing, estimating the amount of wood needed to frame a sloping roof, rendering computer graphics or designing a sewing pattern for the most efficient use of material.

Although there are many types of geometry, school mathematics is devoted primarily to plane Euclidean geometry, studied both synthetically (without coordinates) and analytically (with coordinates). Euclidean geometry is characterized most importantly by the Parallel Postulate, that through a point not on a given line there is exactly one parallel line. (Spherical geometry, in contrast, has no parallel lines.)

During high school, students begin to formalize their geometry experiences from elementary and middle school, using more precise definitions and developing careful proofs. Later in college, some students develop Euclidean and other geometry carefully from a small set of axioms.

The concepts of congruence, similarity, and symmetry can be understood from the perspective of geometric transformation. Fundamentals are the rigid motions: translations, rotations, reflections and combinations of these, all of which are here assumed to preserve distance and angles (and therefore shapes generally). Reflections and rotations each explain a particular type of symmetry and the symmetries of an object offer insight into its attributes—as when the reflective symmetry of an isosceles triangle assures that its base angles are congruent.

In the approach taken here, two geometric figures are defined to be congruent if there is a sequence of rigid motions that carries one onto the other. This is the principle of superposition. For triangles, congruence means the equality of all corresponding pairs of sides and all corresponding pairs of angles. During the middle grades, through experiences drawing triangles from given conditions, students notice ways to specify enough measures in a triangle to ensure that all triangles drawn with those measures are congruent. Once these triangle congruence criteria (ASA, SAS, and SSS) are established using rigid motions, they can be used to prove theorems about triangles, quadrilaterals and other geometric figures.

Similarity transformations (rigid motions followed by dilations) define similarity in the same way that rigid motions define congruence, thereby formalizing the similarity ideas of “same shape” and “scale factor” developed in the middle grades. These transformations lead to the criterion for triangle similarity that two pairs of corresponding angles are congruent.

The definitions of sine, cosine and tangent for acute angles are founded on right triangles and similarity, and, with the Pythagorean Theorem, are fundamental in many real-world and theoretical situations. The Pythagorean Theorem is generalized to non-right triangles by the Law of Cosines. Together, the Laws of Sines and Cosines embody the triangle congruence criteria for the cases where three pieces of information suffice to completely solve a triangle. Furthermore, these laws yield two possible solutions in the ambiguous case, illustrating that Side-Side-Angle is not a congruence criterion.

Analytic geometry connects algebra and geometry, resulting in powerful methods of analysis and problem-solving. Just as the number line associates numbers with locations in one dimension, a pair of perpendicular axes associates pairs of numbers with locations in two dimensions. This correspondence between numerical coordinates and geometric points allows methods from algebra to be applied to geometry and vice versa. The solution set of an equation becomes a geometric curve, making visualization a tool for doing and understanding algebra. Geometric shapes can be described by equations, making algebraic manipulation into a tool for geometric understanding, modeling and proof. Geometric transformations of the graphs of equations correspond to algebraic changes in their equations.

Dynamic geometry environments provide students with experimental and modeling tools that allow them to investigate geometric phenomena in much the same way as computer algebra systems allow them to experiment with algebraic phenomena.

CONNECTIONS TO EQUATIONS

The correspondence between numerical coordinates and geometric points allows methods from algebra to be applied to geometry and vice versa. The solution set of an equation becomes a geometric curve, making visualization a tool for doing and understanding algebra. Geometric shapes can be described by equations, making algebraic manipulation into a tool for geometric understanding, modeling and proof.

Geometry Standards

CONGRUENCE **G.CO**

Experiment with transformations in the plane.

G.CO.2 Represent transformations in the plane using, e.g., transparencies and geometry software; describe transformations as functions that take points in the plane as inputs and give other points as outputs. Compare transformations that preserve distance and angle to those that do not, e.g., translation versus horizontal stretch.

MODELING IN GEOMETRY **G.MG**

Apply geometric concepts in modeling situations.

G.MG.1 Use geometric shapes, their measures, and their properties to describe objects, e.g., modeling a tree trunk or a human torso as a cylinder.★

G.MG.3 Apply geometric methods to solve design problems, e.g., designing an object or structure to satisfy physical constraints or minimize cost; working with typographic grid systems based on ratios.★

High School—Statistics and Probability

Decisions or predictions are often based on data—numbers in context. These decisions or predictions would be easy if the data always sent a clear message, but the message is often obscured by variability. Statistics provides tools for describing variability in data and for making informed decisions that take it into account.

Data are gathered, displayed, summarized, examined and interpreted to discover patterns and deviations from patterns. Quantitative data can be described in terms of key characteristics: measures of shape, center and spread. The shape of a data distribution might be described as symmetric, skewed, flat or bell-shaped, and it might be summarized by a statistic measuring center (such as mean or median) and a statistic measuring spread (such as standard deviation or interquartile range). Different distributions can be compared numerically using these statistics or compared visually using plots. Knowledge of center and spread are not enough to describe a distribution. Which statistics to compare, which plots to use and what the results of a comparison might mean, depends on the question to be investigated and the real-life actions to be taken.

Randomization has two important uses in drawing statistical conclusions. First, collecting data from a random sample of a population makes it possible to draw valid conclusions about the whole population, taking variability into account. Second, randomly assigning individuals to different treatments allows a fair comparison of the effectiveness of those treatments. A statistically significant outcome is one that is unlikely to be due to chance alone, and this can be evaluated only under the condition of randomness. The conditions under which data are collected are important in drawing conclusions from the data; in critically reviewing uses of statistics in public media and other reports, it is important to consider the study design, how the data were gathered and the analyses employed as well as the data summaries and the conclusions drawn.

Random processes can be described mathematically by using a probability model: a list or description of the possible outcomes (the sample space), each of which is assigned a probability. In situations such as flipping a coin, rolling a number cube or drawing a card, it might be reasonable to assume various outcomes are equally likely. In a probability model, sample points represent outcomes and combine to make up events; probabilities of events can be computed by applying the Addition and Multiplication Rules. Interpreting these probabilities relies on an understanding of independence and conditional probability, which can be approached through the analysis of two-way tables.

Technology plays an important role in statistics and probability by making it possible to generate plots, regression functions and correlation coefficients, and to simulate many possible outcomes in a short amount of time.

CONNECTIONS TO FUNCTIONS AND MODELING

Functions may be used to describe data; if the data suggest a linear relationship, the relationship can be modeled with a regression line, and its strength and direction can be expressed through a correlation coefficient.

Statistics and Probability Standards

INTERPRETING CATEGORICAL AND QUANTITATIVE DATA **S.ID**

Summarize, represent and interpret data on a single count or measurement variable.

S.ID.1 Represent data with plots on the real number line (dot plots[G], histograms, and box plots) in the context of real-world applications using the GAISE model.★

MAKING INFERENCES AND JUSTIFYING CONCLUSIONS **S.IC**

Understand and evaluate random processes underlying statistical experiments.

S.IC.2 Decide if a specified model is consistent with results from a given data-generating process, e.g., using simulation. *For example, a model says a spinning coin falls heads up with probability 0.5. Would a result of 5 tails in a row cause you to question the model?*★

CONDITIONAL PROBABILITY AND THE RULES OF PROBABILITY **S.CP**

Understand independence and conditional probability and use them to interpret data.

S.CP.1 Describe events as subsets of a sample space (the set of outcomes) using characteristics (or categories) of the outcomes, or as unions, intersections, or complements of other events (“or,” “and,” “not”).★

Use the rules of probability to compute probabilities of compound events in a uniform probability model.

S.CP.7 Apply the Addition Rule, P(A or B) = P(A) + P(B) − P(A and B), and interpret the answer in terms of the model.★

(+) **S.CP.8** Apply the general Multiplication Rule in a uniform probability model[G], P(A and B) = P(A)·P(B|A) = P(B)·P(A|B), and interpret the answer in terms of the model.★

(+) **S.CP.9** Use permutations and combinations to compute probabilities of compound events and solve problems.★

USING PROBABILITY TO MAKE DECISIONS **S.MD**

Calculate expected values and use them to solve problems.

(+) **S.MD.1** Define a random variable[G] for a quantity of interest by assigning a numerical value to each event in a sample space; graph the corresponding probability distribution[G] using the same graphical displays as for data distributions.★

(+) **S.MD.2** Calculate the expected value[G] of a random variable; interpret it as the mean of the probability distribution.★

Use probability to evaluate outcomes of decisions.

(+) **S.MD.5** Weigh the possible outcomes of a decision by assigning probabilities to payoff values and finding expected values.★

- **a.** Find the expected payoff for a game of chance. For example, find the expected winnings from a state lottery ticket or a game at a fast-food restaurant.
- **b.** Evaluate and compare strategies on the basis of expected values. For example, compare a high-deductible versus a low-deductible automobile insurance policy using various, but reasonable, chances of having a minor or a major accident.

(+) **S.MD.6** Use probabilities to make fair decisions, e.g., drawing by lots, using a random number generator.★

(+) **S.MD.7** Analyze decisions and strategies using probability concepts, e.g., product testing, medical testing, pulling a hockey goalie at the end of a game.★

Computer Science Grades 9-12—Foundational Level

NETWORKS AND THE INTERNET

Networking

NI.N.9-12.F.c Understand scalability and reliability of networks to describe the relationships and effects of how the different types of networks work together.

Cybersecurity

NI.C.9-12.F.a Examine and employ principles of cybersecurity.
NI.C.9-12.F.b Identify physical, social and digital security risks to address possible attacks.
NI.C.9-12.F.d Explore and utilize examples of encryption methods, e.g., Vigenere, Bacon’s cipher and Enigma

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.9-12.F.a Analyze patterns in a real-world data store through hypothesis, testing and use of data tools to gain insight and knowledge.

Visualization and Communication

DA.VC.9-12.F.a Analyze the benefits and limitations of data visualization or multisensory artifacts and tools to communicate which is most appropriate to solve a real-world problem.

Inference and Modeling

DA.IM.9-12.F.a Evaluate a model by creating a hypothesis, testing it and refining it to discover connections and trends in the data.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.9-12.F.a Define and use appropriate problem solving strategies and visual artifacts to create and refine a solution to a real-world problem.
ATP.A.9-12.F.b Define and implement an algorithm by decomposing problem requirements from a problem statement to solve a problem.
ATP.A.9-12.F.c Define and explain iterative algorithms to understand how and when to apply them.

ATP.A.9-12.F.d Define and explain recursive algorithms to understand how and when to apply them.

Variables and Data Representation

ATP.VDR.9-12.F.a Identify types of variables and data and utilize them to create a computer program that stores data in appropriate ways.

Control Structures

ATP.CS.9-12.F.a Define control structures and Boolean logic and use them to solve real-world scenarios.
ATP.CS.9-12.F.b Use appropriate syntax to create and use a method.

Modularity

ATP.M.9-12.F.a Break down a solution into procedures using systematic analysis and design
ATP.M.9-12.F.b Create computational artifacts by systematically organizing, manipulating and/or processing data.

ARTIFICIAL INTELLIGENCE

Perception

AI.RR.9-12.F.a Categorize real-world problems as classification, prediction, sequential decision problems, combination search, heuristic search, adversarial search, logical deduction or statistical inference.

IMPACTS OF COMPUTING

Social Interactions

IC.SI.9-12.F.a Evaluate tools to increase connectivity of people in different cultures and career fields.

Safety, Law and Ethics

IC.SLE.9-12.F.b Analyze the concepts of usability and security to explain typical tradeoffs between them.

Computer Science Grades 9-12—Advanced Level

NETWORKS AND THE INTERNET

Cybersecurity

NI.C.9-12.A.d Explore and utilize examples of encryption methods (e.g., Vigenére, Bacon’s cipher and Enigma).

DATA AND ANALYSIS

Visualization and Communication

DA.VC.9-12.A.a Create visualization or multisensory artifacts to communicate insights and knowledge gained from complex data analysis that answers real-world questions.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.9-12.A.a Define and explain recursive algorithms to understand how and when to apply them.
ATP.A.9-12.A.b Use iteration to effectively solve problems.
ATP.A.9-12.A.c Use recursion to effectively solve problems.

Variables and Data Representation

ATP.VDR.9-12.A.a Utilize different data storage structures to store larger and more complex data than variables can contain.
ATP.VDR.9-12.A.b Identify the appropriate data structures or variables to use to design a solution to a complex problem.

Control Structures

ATP.CS.9-12.A.a Write programs that use library methods and control structures and methods to solve a problem.
ATP.CS.9-12.A.b Refactor a program to be smaller and more efficient.

Modularity

ATP.M.9-12.A.a Construct solutions to problems using student-created components (e.g., procedures, modules, objects).

ARTIFICIAL INTELLIGENCE

Representation & Reasoning

AI.RR.9-12.A.b Illustrate breadth-first, depth-first and best-first search algorithms to grow a search tree.

IMPACTS OF COMPUTING

Safety, Law and Ethics

IC.SLE.9-12.A.a Create a scenario to demonstrate typical tradeoffs between usability and security and recommend security measures based on these or other tradeoffs.

Glossary

Bivariate data. Pairs of linked numerical observations. Example: a list of heights and weights for each player on a football team.

Box plot. A method of visually displaying a distribution of data values by using the median, quartiles, and extremes of the data set. A box shows the middle 50% of the data.[1]
See also: first quartile and third quartile.

Expected value. For a random variable, the weighted average of its possible values, with weights given by their respective probabilities.

First quartile. For a data set with median M, the first quartile is the median of the data values less than M. Example: For the data set {1, 3, 6, 7, 10, 12, 14, 15, 22, 120}, the first quartile is 6.2
See also: median, third quartile, interquartile range.

GAISE Model *See also:* Pre-K-12 Guidelines for Assessment and Instruction in Statistics Education II (GAISE II): A Framework for Statistics and Data Science Education

Independently combined probability models. Two probability models are said to be combined independently if the probability of each ordered pair in the combined model equals the product of the original probabilities of the two individual outcomes in the ordered pair.

Interquartile Range. A measure of variation in a set of numerical data, the interquartile range is the distance between the first and third quartiles of the data set. Example: For the data set {1, 3, 6, 7, 10, 12, 14, 15, 22, 120}, the interquartile range is 15 − 6 = 9.
See also: first quartile, third quartile.

Justify: To provide a convincing argument for the truth of a statement to a particular audience.

Line plot. A method of visually displaying a distribution of data values where each data value is shown as a dot or mark above a number line. Also known as a dot plot.[3]

Matrix (plural 'matrices') A collection of numbers, symbols, expressions, or images arranged in a grid (rows and columns) to form a rectangular array. In mathematics, it is used to represent transformations or objects. In computer science, it may be used to represent a group of related data.

Mean. A measure of center in a set of numerical data, computed by adding the values in a list and then dividing by the number of values in the list. (To be more precise, this defines the arithmetic mean) Example: For the data set {1, 3, 6, 7, 10, 12, 14, 15, 22, 120}, the mean is 21.

Median. A measure of center in a set of numerical data. The median of a list of values is the value appearing at the center of a sorted version of the list—or the mean of the two central values if the list contains an even number of values. Example: For the data set {2, 3, 6, 7, 10, 12, 14, 15, 22, 90}, the median is 11.

Mean absolute deviation. A measure of variation in a set of numerical data, computed by adding the distances between each data value and the mean, then dividing by the number of data values. Example: For the data set {2, 3, 6, 7, 10, 12, 14, 15, 22, 120}, the mean absolute deviation is 20.

---

1. Adapted from Wisconsin Department of Public Instruction, http://dpi.wi.gov/standards/mathglos.html, accessed March 2, 2010.

2. Many different methods for computing quartiles are in use. The method defined here is sometimes called the Moore and McCabe method. See Langford, E., "Quartiles in Elementary Statistics," Journal of Statistics Education Volume 14, Number 3 (2006).

3. Adapted from Wisconsin Department of Public Instruction, *op. cit.*

4. Adapted from Wisconsin Department of Public Instruction, *op. cit.*

Participatory Sensing. An approach to data collection and interpretation in which individuals, acting alone or in groups, use their personal mobile devices and web services to systematically explore interesting aspects of their worlds, ranging from health to culture.

Pre-K-12 Guidelines for Assessment and Instruction in Statistics Education II (GAISE II): A Framework for Statistics and Data Science Education. It is an updated report endorsed by the American Statistical Association (ASA) and the National Council of Teachers of Mathematics (NCTM) to enhance the Statistics standards. Like the GAISE I, it provides a framework of recommendations for developing students' foundational skills in statistical reasoning in three levels across the school years, described as levels A, B, and C. GAISE I and GAISE II can be found on the American Statistical Association website.

Probability distribution. The set of possible values of a random variable with a probability assigned to each.

Probability. A number between 0 and 1 is used to quantify the likelihood of processes that have uncertain outcomes (such as tossing a coin, selecting a person at random from a group of people, tossing a ball at a target, or testing for a medical condition).

Probability model. A probability model is used to assign probabilities to outcomes of a chance process by examining the nature of the process. The set of all outcomes is called the sample space, and their probabilities sum to 1. *See also:* uniform probability model.

Prove: To provide a logical argument that demonstrates the truth of a statement. A proof is typically composed of a series of justifications, which are often single sentences and may be presented informally or formally.

Random variable. An assignment of a numerical value to each outcome in a sample space.

Sample space. In a probability model for a random process, a list of the individual outcomes that are to be considered.

Scatter plot. A graph in the coordinate plane represents a set of bivariate data. For example, the heights and weights of a group of people could be displayed on a scatter plot.[4]

Third quartile. For a data set with median *M*, the third quartile is the median of the data values greater than *M*. Example: For the data set {2, 3, 6, 7, 10, 12, 14, 15, 22, 120}, the third quartile is 15.
See also: median, first quartile, interquartile range.

Uniform probability model. A probability model which assigns equal probability to all outcomes. *See also:* probability model.

Verify: To check the truth or correctness of a statement in specific cases

Computer Science References

Bureau of Labor Statistics. (2015). Labor force statistics from the Current Population Survey [Data file]. Retrieved from http://www.bls.gov/cps

Change the Equation. (2016, August 9). New data: Bridging the computer science access gap [Blog post]. Retrieved from http://changetheequation.org/blog/new-data-bridging-computer-science-access-gap-0 (no longer available)

College Board. (2016). AP program participation and performance data 2015 [Data file]. Retrieved from https://research.collegeboard.org/programs/ap/data/archived/ap-2015

Denner, J., Martinez, J., Thiry, H., & Adams, J. (2015). Computer science and fairness: Integrating a social justice perspective into an after-school program. Science Education and Civic Engagement: An International Journal, 6(2): 41–54.

Goode, J. (2008, March). Increasing diversity in K–12 computer science: Strategies from the field. ACM SIGCSE Bulletin, 40(1), 362–366.

Hansen, A., Hansen, E., Dwyer, H., Harlow, D., & Franklin, D. (2016). Differentiating for diversity: Using universal design for learning in computer science education. Proceedings of the 47th ACM Technical Symposium on Computing Science Education (pp. 376–381).

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org

Kumar, A. N. (2012, July). A study of stereotype threat in computer science. Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (pp. 273–278).

Margolis, J., Ryoo, J., Sandoval, C., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. ACM Inroads, 3(4), 72–78.

Margolis, J., Goode, J., Chapman, G., & Ryoo, J. J. (2014). That classroom 'magic.' Communications of the ACM, 57(7), 31–33.

Morgan, R., & Klaric, J. (2007). AP students in college: An analysis of five-year academic careers. Research report no. 2007-4. Retrieved from https://eric.ed.gov/?id=ED561034

Quorum [Computer software]. (2019). Retrieved from https://quorumlanguage.com/

Snodgrass, M. R., Israel, M., & Reese, G. (2016). Instructional supports for students with disabilities in K–5 computing: Findings from a cross-case analysis. Computers & Education, 100, 1–17.

Strawhacker, A. L., & Bers, M. U. (2014, August). ScratchJr: Computer programming in early childhood education as a pathway to academic readiness and success. Poster presented at DR K–12 PI Meeting, Washington, DC

Acknowledgements

MATH STANDARDS ADVISORY COMMITTEE MEMBERS

Aaron Altose
The Ohio Mathematics Association of Two-Year Colleges

Jeremy Beardmore
Ohio Educational Service Center Association

Jessica Burchett
Ohio Teachers of English to Speakers of Other Languages

Jeanne Cerniglia
Ohio Education Association

Margie Coleman
Cochair

Jason Feldner
Ohio Association for Career and Technical Education

Brad Findell
Ohio Higher Education

Gregory D. Foley
Ohio Mathematics and Science Coalition

Margaret (Peggy) Kasten
Cochair

Courtney Koestler
Ohio Mathematics Education Leadership Council

Scott Mitter
Ohio Math and Science Supervisors

Tabatha Nadolny
Ohio Federation of Teachers

Eydie Schilling
Ohio Association for Supervision and Curriculum Development

Kim Yoak
Ohio Council of Teachers of Mathematics

MATH STANDARDS WORKING GROUP MEMBERS

Ali Fleming
Teacher, Bexley City, C

Gary Herman
Curriculum Specialist/Coordinator, Putnam County ESC, NW

William Husen
Higher Education, Ohio State University, C

Kristen Kelly
Curriculum Specialist/Coordinator, Cleveland Metropolitan School District, NE

Endora Kight Neal
Curriculum Specialist/Coordinator, Cleveland Metropolitan School District, NE

Dawn Machacek
Teacher, Toledo Public Schools, NW

Sherryl Proctor
Teacher, Vantage Career Center, NW

Tess Rivero
Teacher, Bellbrook-Sugarcreek Schools, SW

Jennifer Walls
Teacher, Akron Public Schools, NE

Gaynell Wamer
Teacher, Toledo City, NW

Sandra Wilder
Teacher, Akron Public Schools, N

COMPUTER SCIENCE ADVISORY COMMITTEE MEMBERS (2022)

Eva Bradshaw
National Center for Women and Information Technology

Carmen Bryson
East Cleveland City Schools

Crystal Franklin
Cleveland State University

Kelly Gaier Evans
Battelle-Ohio STEM Learning Network

Stephanie Hoeppner
Live Oaks Career Campus

Jim Johnson
Morgan Local Schools
CSTA Ohio

Ryan Johnston
C-TEC of Licking County

Megan Kinsey
Apple

Chelsey Cook-Kohn
Cleveland Metropolitan School District
Cleveland State University

Alex Kotran
The AI Education Project

Mike Lawson
Morgan Junior High School

Cristina Nowak
The AI Education Project

Krystina Pratt
Teaching & Learning Collaborative

Jason Scherer
Field Local, OEA

Kelly Shrewsberry
Teaching & Learning Collaborative

Stanley Smith
The University of Akron

Michelle Snow
C-TEC

Randy Tucker
Tech Corps

COMPUTER SCIENCE WORKING GROUP MEMBERS (2022)

Jennifer Blackledge
Ohio Council for Teachers of Mathematics

Tracy Bronner
Clermont Northeastern Local

Betty Cantley
Retired teacher

Elizabeth Curtis
Cleveland Metropolitan

Karen Bergreen
Logan Hocking Local

Elizabeth Davis
LexisNexis

Roberta Bandfield
Corpus Christi Academy

John Davis III
Greater Cleveland Partnership/OHTec

Bryan Drost
Rocky River City

Ray Gaier
Kings Local

Gary Herman
Ohio Chamber of Commerce

Shanshan Huang
Educational Solutions Co.

Krissy Machamer
Licking Heights Local

Jennifer Nichols
Computer Science Teachers Association

Amanda O'Mara
Microsoft TEALS

Tom O'Neill
Butler Tech

Tammi Ramsey
Washington Court House City

Darrin Spondike
Auburn Career Center

James Stanton
Battelle Education

Sarah Thornton
Pickaway-Ross Career and Technology Center/Zane Trace High School

Courtney Webb
Euclid City Schools

Vicki Willett
Licking Heights Local

Matthew Williams
Avon Lake City

Matthew Yuhasz
Columbus City

COMPUTER SCIENCE WRITING TEAM MEMBERS

Julie Hagaman
Tri-County North School District

Gary Herman
Putnam County ESC

Marvin "Mel" Hoffert
Sycamore Community

Jennifer Humphrey
Cleveland Metropolitan

Diane Kahle
Upper Arlington City

Christopher Kaminski
Fairview Park City

Barbara Kenne
West Clermont Local

Bonnie Kovacic
Perry Local

Carlo Miraldi
Great Oaks Career Campuses

Valerie Moga
Hudson City

Shellu Muma
Ottoville Local

Pat Murakami
Dayton Regional STEM School

Lisa Murray
Vermilion Local

Lynne Pachnowski
University of Akron

Erin Pekar
Maple Heights City

Krystina Pratt
New Albany Plain Local

Karen Plaster
University of Akron

Leslie Slaven
Lakota Local

Renee Snyder
Teaching & Learning Collaborative

Darrin Spondike
Auburn City

Rebecca Stanek
Stow-Munroe Falls City

James Stanton
Maumee City

Deborah Wiley
North Canton City Schools
Kent State Stark University

Jonathon Wilson
New Richmond Exempted Village

Robert Wyant
Austintown Local

OHIO HIGH SCHOOL MATH PATHWAYS ADVISORY COUNCIL

Trina Barrell
Buckeye Association of School Administrators (BASA)

Michael Broadwater
Ohio Association of Secondary School Administrators (OASSA)

Mark Cortez
Higher education admissions

Kevin Duff
Ohio Excels

Shawn Grime
Ohio School Counselor Association (OSCA)

Kelly Hogan
Ohio Association of Community Colleges (OACC)

Aaron Johnson
Ohio Association for Career and Technical Education (Ohio ACTE)

Tom Kaczmarek
Ohio Education Association (OEA)

Peggy Kasten
Ohio Mathematics and Science Coalition (OMSC)

Todd Martin
The Ohio 8 Coalition

Ricardo Moena
Ohio Mathematics Initiative (OMI)

Carrie Rice
Ohio Federation of Teachers (OFT)

Brad Ritchey
Ohio School Boards Association (OSBA)/ Buckeye Association of School Administrators (BASA)

Barbara Varley
Ohio Parent Teacher Association (Ohio PTA)

Heather Wukelich
Ohio Council of Teachers of Mathematics (OCTM)

OHIO HIGH SCHOOL MATH PATHWAYS ARCHITECTS

Deborah Ackley
Teacher, Toledo Public Schools, NW

Dave Burkhart
Teacher, New Lexington City Schools, SE

Doug Darbro
Higher Education, Shawnee State University, SE

Brad Findell
Higher Education, The Ohio State University, C

Derek Gulling
Teacher, Edison Local Schools (Jefferson County), NE

Christina Hamman
Curriculum Specialist/Coordinator, Medina City Schools, NE

Gary Herman
Curriculum Specialist/Coordinator, Putnam County ESC, NW

Ruth Hopkins
Teacher, Symmes Valley Local Schools, SE

Endora Kight Neal
Higher Education, Cuyahoga Community College and Curriculum Specialist/Coordinator, Cleveland Metropolitan School District, NE

Scott Mitter
Teacher, Kettering City Schools, SE

Jennifer Montgomery
Teacher, Wayne Local Schools, SW

Rachael Newell
Teacher, Perrysburg Exempted Village, NW

Stephanie Stafford
Higher Education, Cincinnati State Technical and Community College, SW

Lee Wayand
Higher Education, Columbus State Community College, C

Richelle Zbinden
Teacher, Miami Valley Career Technology Center, SW

OHIO HIGH SCHOOL DISCRETE MATHEMATICS/COMPUTER SCIENCE WORKGROUP

Katherine Blazyk
Teacher, Beavercreek City Schools

Jennifer Boughton
Teacher, Northwest Local Schools

Frank Carraher
Curriculum Specialist/Coordinator, ESC of the Western Reserve

Bryan Drost
Curriculum Specialist/Coordinator, Rocky River Schools

Carole Katz
Curriculum Specialist/Coordinator, Beachwood City Schools

Andrew King
Teacher, Mississinawa Valley Schools

Christopher Orban
Higher Education, The Ohio State University

Doug Roberts
Teacher, Metro Early College High School

James Stanton
Teacher, Maumee City Schools

Jeff VanArnhem
Teacher, Olmsted Falls City

Carol VanFossen
Teacher, Metro Early College High School

Lee Wayand
Higher Education, Columbus State Community College