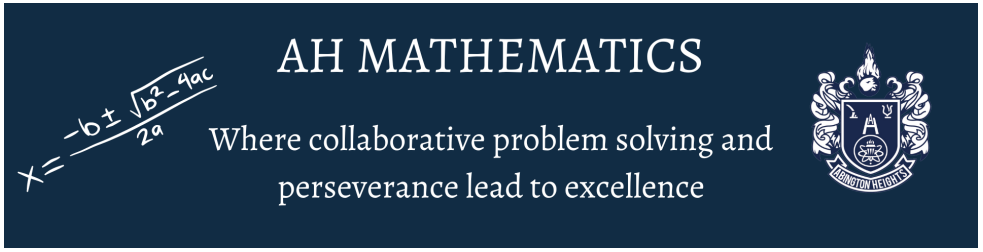
Abington Heights School District
Computer Science Principles L2 Curriculum

In Computer Science Principles L2, students develop their computer programming skills through the following areas of study:

1. Creative Development
2. Data
3. Algorithms and Programming
4. Computer Systems and Networks
5. Impact of Computing

Board Approval Date: June 7, 2023
Adoption: 2023 - 2024 SY
Review Date:



Abington Heights Math Framework

Stakeholders	Actions
Students	★ Engage in mathematical discussions, share their ideas openly, be inquisitive, seek to understand and learn more about mathematical concepts, and try their best daily. ★ Exhibit creativity and curiosity in problem solving individually and collaboratively. ★ Persevere in engaging and challenging daily mathematical practice. ★ Come prepared to learn every day.
Teachers	★ Create a safe and collaborative classroom environment where students feel vested in a shared vision for mathematical excellence. ★ Develop high quality instruction that meets the needs of all learners through differentiation. ★ Use a variety of 21st century methodologies to advance learning. ★ Partner with parents and guardians to support student success. ★ Establish a collaborative community within the building and amongst grade levels to ensure a cohesive level of instruction.
Building Leaders	★ Deeply understand the needs of teachers, students, the instructional materials being used, programs being implemented, and the expectations for state-level assessment scores ○ Knowledgeable about program and grade level standards ○ Ensure consistent and equal access to high-quality instructional materials and resources, building. ★ Be partners with teachers, students and families: ○ Provide guidance and support to the mathematical community. ○ Understand needs of teachers, students and families. ★ Trust the educators to make professional decisions based on program, student, and district needs.
Central Admin	★ Effectively communicate to the school board and community specific areas of need and how to support teachers and building leaders in a quest for mathematical excellence ★ Deeply understand the needs of teachers, students, the instructional materials being used, programs being implemented, and the expectations for state-level assessment scores ○ Have a common metric for mathematical excellence. ○ Ensure consistent and equal access to high-quality instructional materials and resources, district. ○ Re-examine best practices/curriculum routinely (6 years). ★ Support a culture of collaboration between the other stakeholder groups to maintain the standard of excellence of the Abington Heights ★ Trust the educators to make professional decisions based on program, student, and district needs.
Parents/ Community	★ Be a strong support system and contribute by building a positive math community for students. ★ Encourage a positive math mindset. ★ Have conversations with their children about school and ask what they are learning about in school. ★ Be open, receptive to the district's ideas about student learning and reach out to teachers/school to learn more about how they can support. ★ Trust the educators to make professional decisions based on program, student, and district needs.
School Board	★ Provide the fiscal resources to support: ○ Highly qualified professionals for mathematics ○ High-quality instructional materials ○ Effective and efficient math interventions for remediation ○ Professional development for math content and instructional practices ★ Trust the educators to make professional decisions based on program, student, and district needs.

Computer Science Principles L2 Scope and Sequence

Month	Unit	Estimated Number of Weeks
September	Bits and Bytes	3
	Networks and the Internet	1
October	Networks and the Internet	3
	Global Impacts	1
November	Global Impacts	2
	Data and Privacy (Cybersecurity)	2
December	Data and Privacy (Cybersecurity)	1
	Introduction to Programming	2
January	Introduction to Programming	2
	Logic Operators	2
February	Logic Operators	1
	Control Structures	3
March	Control Structures	2
	Lists	2
April	Lists	2
	Functions	2
May	Functions	2
	Final Project	2
June	Final Project	1

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Bits and Bytes	How is digital data stored, compressed, and manipulated on digital devices? How does a digital device make sense of bits and bytes? How are graphics and images stored and compressed?	Digital Data Bits and Bytes Abstraction Binary Conversions Digital vs Analog Compression Lossy vs Lossless	Explain how data can be represented using bits. Explain the consequences of using bits to represent data. Calculate and compare binary (base 2) numbers to decimal (base 10) and interpret the number of bits Explain how computing innovations are improved through collaboration.	Digital Information Lecture Binary Secret Message Assignment Worksheet Various Youtube Educational Videos (In Lecture)	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz Kahoot Review Sheet Test
Networks and the Internet	How does the internet and networks actually work?	Performance Protocols Client and Server Routers and Packets Domain Name System Search Engines	Explain how computing devices work together in a network. Explain (in detail) how the internet works. Explain how data is sent through the internet via packets. Describe the differences between the Internet and the World Wide Web.	Internet and Network Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz Kahoot Test

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Networks and the Internet (continued)			Research the benefits of fault-tolerance and fault-tolerant systems. Explain redundancy.		
Global Impacts	How has computing changed our world? What are the benefits and downfalls of computing and computing innovations? What are the legal and ethical implications of computing?	Computing Changing the World Benefits of Computing Downfalls of Computing Legal and Ethical Implications of Computing	Explain how an effect of computing innovation can be both beneficial and harmful. Explain how a computing innovation can have an impact beyond its intended purpose. Describe issues that contribute to the digital divide. Explain how bias exists in computing innovations. Explain how people participate in problem solving processes at scale. Explain how the use of computing can raise legal and ethical concerns.	Impacts of Computing Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz Test

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Global Impacts (continued)			Describe the risks to privacy from collecting and storing personal data on a computer system.		
Data and Privacy (Cybersecurity)	How does the use of computer programs hurt/help online digital privacy? How can using computers involve risks to personal safety and identity? How unauthorized access to computing resources is gained?	Data and Metadata Processing Data Protecting Data Misusing Data Phishing/Keylogging/etc	Describe what kind of information can be extracted from data/metadata. Explain how programs can be used to gain insight and knowledge from data. Explain how unauthorized access to computing resources is gained.	Data and Privacy Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz Test
Introduction to Programming	How do we initiate the programming process? Why learn a programming language?	Introduction to Programming Output Data Types, Strings and Variables String Concatenation Input Comments in Python Debugging in Python	How do we use syntax rules in the Python Programming Language? How to represent a value with a variable. Determine the value of a variable as a result of an assignment. How and when to write a procedure and call it.	Intro to Programming Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz I do- We do- You do Programming Practice Programming Lab Test

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Logic Operators	What is a mathematical operator and when should we use them in programming? How do we understand how a program makes a decision?	Math Operators Modulus Order of Operations Comparison Operators Logical Operators Using Random()	Evaluate expressions that use arithmetic operators. Evaluate expressions that manipulate strings. Determine relationships between two variables, expressions, or values. Determine truth table values for boolean expressions. Generate random values using an algorithm and formula written in code.	Math Operators/Logic Operators Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz I do- We do- You do Programming Practice Programming Lab Test
Control Structures	Why do computer scientists use algorithms and control structures? Why are algorithms important in areas other than math/computer science? How do algorithms and control structures help us in everyday life?	Algorithm Basics and Tracing Control Structures -Sequences Control Structures - Iteration Iterations in Python Control Structures - Selections Selections in Python	Express an algorithm that uses sequencing without using a programming language. Represent a step-by-step algorithmic process using sequential code statements.	Algorithm and Control Structures Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz I do- We do- You do Programming Practice Programming Lab Test

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Control Structures (continued)		Nesting Iteration and Selection	Express an algorithm that uses selection without using a programming language. Write and Trace conditional statements and determine the results of pre-written conditional statements in Python. Express an algorithm that uses iteration without using a programming language. Write and Trace iteration statements and determine the results of pre-written iteration statements in Python. Compare multiple algorithms to determine if they yield the same results and determine which is most efficient. Create and combine algorithms that can help solve a problem.		

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Lists	Why are lists an effective way in programming to organize and interpret data? What is the difference between a list and a database?	Lists Traversing and Searching Lists Adding to Lists Strings with Lists Multi-dimensional Lists	Represent a list or string using a variable. Develop data abstraction using lists to store multiple elements. Explain how the use of data abstractions manages complexity in program code. Write expressions that use list indexing and list procedures. Evaluate expressions that use list indexing and list procedures. Write iteration statements to traverse a list. Determine the result of an algorithm that includes list traversals.	Lists and Data Lecture Worksheet Kahoot Various Youtube Educational Videos (Homework) CS Podcast	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz I do- We do- You do Programming Practice Programming Lab Test
Functions	Why are functions used in programming? What is the proper syntax to write a basic function in Python?	Function Basics Lists in Functions Returning Values from a Function Varying Parameters in a Function	Write statements to call procedures. Determine the result or effect of a procedure call.	Functions and Procedures Lecture Worksheet Kahoot	Scaffolded Questions Mid Lecture Checkpoint (Understanding) Questions Quiz

	Essential Questions	Content	Skills	Activities	Assessment/Evidence of Learning
Functions (continued)			Trace the route a procedure call takes while running Python Code. Explain how the use of procedural abstraction manages complexity in a program. Develop procedural abstractions to manage complexity in a program by writing procedures in the Python language.	Various Youtube Educational Videos (Homework) CS Podcast	I do- We do- You do Programming Practice Programming Lab Test
Final Project	N/A			Final Project	Final Project