

Computer Science I

Computer Science Pre-AP

At-A-Glance - Lamar CISD

Ongoing Skills Imbedded All Year	Professional Standards/Employability Skills/Technical Skills		
	2(D) The student will write programs with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, spacing, indentation, and a standardized program style. 4(H) The student will identify and debug errors. 4(P) The student will create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root. 4(R) The student will develop sequential algorithms to solve non-branching and non-iterative problems. 6(T) The student will identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data		
Grading Period	Unit Name	Estimated Time Frame	TEKS
Grading Period 1 29 Days	Introductory Skills/Set Up	3 Days	2H
	2(H) The student will seek and respond to advice from peers and professionals in evaluating quality and accuracy.		
	Unit 1: Welcome	2 Days	6A, 6B, 6C, 6D, 6V
	6(A) The student will compare and contrast types of operating systems, software applications, and programming languages. 6(B) The student will demonstrate knowledge of major hardware components, including primary and secondary memory, a central processing unit (CPU), and peripherals. 6(C) The student will differentiate among current programming languages, discuss the use of those languages in other fields of study, and demonstrate knowledge of specific programming terminology and concepts. 6(D) The student will differentiate between a high-level compiled language and an interpreted language. 6(V) The student will compare and contrast strongly typed and un-typed programming languages.		
	Unit 2: Basic Python and Console Interaction	14 Days	2A, 2B, 2D, 2R, 4D, 4F, 4H, 4I, 4J, 4O, 4Q, 4R, 6O, 6P, 6R
2(A) The student will create and properly display meaningful output. 2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user. 2(D) The student will write programs and communicate with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, indentation, and a standardized program style. 2(R) The student will develop sequential algorithms to solve non-branching and non-iterative problems. 4(D) The student will identify the data types and objects needed to solve a problem. 4(F) The student will design a solution to a problem. 4(H) The student will identify and debug errors. 4(I) The student will test program solutions with appropriate valid and invalid test data for correctness. 4(J) The student will debug and solve problems using error messages, reference materials, language documentation, and effective strategies. 4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division. 4(Q) The student will develop program solutions that use assignment. 4(R) The student will develop sequential algorithms to solve non-branching and non-iterative problems. 6(O) The student will choose, identify, and use the appropriate data types for integer, real, and Boolean data when writing program solutions. 6(P) The student will demonstrate an understanding of the concept of a variable. 6(R) The student will demonstrate an understanding of how to represent and manipulate text data, including concatenation and other string functions.			

	Unit 3: Introduction to Programming with Turtle Graphics	10 Days	2A, 2B, 2D, 2F, 4A, 4B, 4C, 4D, 4E, 4F, 4G, 4I, 4N, 4O, 4Q, 4R, 4S, 4T, 6H, 6P
Grading Period 2 26 Days	Unit 3: Introduction to Programming with Turtle Graphics continues	14 Days	2A, 2B, 2D, 2F, 4A, 4B, 4C, 4D, 4E, 4F, 4G, 4I, 4N, 4O, 4Q, 4R, 4S, 4T, 6H, 6P
<p>2(A) The student will create and properly display meaningful output.</p> <p>2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user.</p> <p>2(D) The student will write programs and communicate with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, indentation, and a standardized program style.</p> <p>2(F) The student will display simple vector graphics using lines, circles, and rectangles.</p> <p>4(A) The student will use program design problem-solving strategies to create program solutions.</p> <p>4(B) The student will define and specify the purpose and goals of solving a problem.</p> <p>4(C) The student will identify the subtasks needed to solve a problem.</p> <p>4(D) The student will identify the data types and objects needed to solve a problem.</p> <p>4(E) The student will identify reusable components from existing code.</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(G) The student will code a solution from a program design.</p> <p>4(I) The student will analyze and modify existing code to improve the underlying algorithm.</p> <p>4(N) The student will select the most appropriate algorithm for a defined problem.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(R) The student will develop sequential algorithms to solve non-branching and non-iterative problems.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(T) The student will develop iterative algorithms and code programs to solve practical problems.</p> <p>6(H) The student will create subroutines that do not return values with and without the use of arguments and parameters.</p> <p>6(P) The student will demonstrate an understanding of the concept of a variable.</p>			

	Unit 4: Conditionals	12 Days	2A, 2B, 4F, 4H, 4I, 4J, 4O, 4P, 4Q, 4S, 4U, 4V
Grading Period 3 25 Days	Unit 4: Conditionals	12 Days	2A, 2B, 4F, 4H, 4I, 4J, 4O, 4P, 4Q, 4S, 4U, 4V
			<p>2(A) The student will create and properly display meaningful output.</p> <p>2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user.</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(H) The student will identify and debug errors.</p> <p>4(I) The student will test program solutions with appropriate valid and invalid test data for correctness.</p> <p>4(J) The student will debug and solve problems using error messages, reference materials, language documentation, and effective strategies.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(P) The student will create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(U) The student will demonstrate proficiency in the use of the relational operators.</p> <p>4(V) The student will demonstrate proficiency in the use of the logical operators.</p>
	Unit 5: Looping	13 Days	2A, 2B, 4F, 4I, 4K, 4L, 4N, 4O, 4P, 4Q, 4S, 4T
		<p>2(A) The student will create and properly display meaningful output.</p> <p>2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user.</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(I) The student will analyze and modify existing code to improve the underlying algorithm.</p> <p>4(K) The student will explore common algorithms, including finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average.</p> <p>4(L) The student will analyze and modify existing code to improve the underlying algorithm.</p> <p>4(N) The student will select the most appropriate algorithm for a defined problem.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(P) The student will create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(T) The student will develop iterative algorithms and code programs to solve practical problems.</p>	

	Unit 6: Functions and Exceptions	16 Days	2A, 2B, 4E, 4F, 4H, 4I, 4L, 4M, 4O, 4Q, 4S, 4T, 6F, 6H, 6I, 6S
	<p>2(A) The student will create and properly display meaningful output.</p> <p>2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user.</p> <p>4(E) The student will identify reusable components from existing code</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(H) The student will identify and debug errors.</p> <p>4(I) The student will test program solutions with appropriate valid and invalid test data for correctness.</p> <p>4(L) The student will analyze and modify existing code to improve the underlying algorithm.</p> <p>4(M) The student will create program solutions that exhibit robust behavior by understanding, avoiding, and preventing runtime errors, including division by zero and type mismatch.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(T) The student will develop iterative algorithms and code programs to solve practical problems.</p> <p>6(F) The student will use local and global scope access variable declarations.</p> <p>6(H) The student will create subroutines that do not return values with and without the use of arguments and parameters.</p> <p>6(I) The student will create subroutines that return typed values with and without the use of arguments and parameters.</p> <p>6(S) The student will demonstrate an understanding of the concept of scope.</p>		
Grading Period 4 32 Days	Unit 7: Strings	16 Days	2A, 2B, 4F, 4H, 4I, 4J, 4O, 4Q, 4S, 4T, 6R
	<p>2(A) The student will create and properly display meaningful output.</p> <p>2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user.</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(H) The student will identify and debug errors.</p> <p>4(I) The student will test program solutions with appropriate valid and invalid test data for correctness.</p> <p>4(J) The student will debug and solve problems using error messages, reference materials, language documentation, and effective strategies.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(T) The student will develop iterative algorithms and code programs to solve practical problems.</p> <p>6(R) The student will demonstrate an understanding of how to represent and manipulate text data, including concatenation and other string functions.</p>		
Grading Period 5 32 Days	Unit 8: Creating and Altering Data Structures	16 Days	2A, 2B, 4D, 4F, 4I, 4K, 4O, 4P, 4Q, 4S, 4T, 6T, 6U
	<p>2(A) The student will create and properly display meaningful output.</p> <p>2(B) The student will create interactive console display interfaces, with appropriate user prompts, to acquire data from a user.</p> <p>4(D) The student will identify the data types and objects needed to solve a problem.</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(I) The student will analyze and modify existing code to improve the underlying algorithm.</p> <p>4(K) The student will explore common algorithms, including finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(P) The student will create program solutions to problems using available mathematics libraries, including absolute value, round, power, square, and square root.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(T) The student will develop iterative algorithms and code programs to solve practical problems.</p>		

	<p>6(T) The student will identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data.</p> <p>6(U) The student will choose, identify, and use the appropriate data type and structure to properly represent the data in a program problem solution.</p>		
	Unit 9: Extending Data Structure.	16 Days	2A, 2D, 4F, 4G, 4O, 4Q, 4R, 4S, 4T, 6U
	<p>2(A) The student will create and properly display meaningful output.</p> <p>2(D) The student will write programs and communicate with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, indentation, and a standardized program style.</p> <p>4(F) The student will design a solution to a problem.</p> <p>4(G) The student will code a solution from a program design.</p> <p>4(O) The student will demonstrate proficiency in the use of the arithmetic operators to create mathematical expressions, including addition, subtraction, multiplication, real division, integer division, and modulus division.</p> <p>4(Q) The student will develop program solutions that use assignment.</p> <p>4(R) The student will develop sequential algorithms to solve non-branching and non-iterative problems.</p> <p>4(S) The student will develop algorithms to decision-making problems using branching control statements.</p> <p>4(T) The student will develop iterative algorithms and code programs to solve practical problems.</p> <p>6(U) The student will choose, identify, and use the appropriate data type and structure to properly represent the data in a program problem solution.</p>		
Grading Period 6 29 Days	End of the year project	28 Days	4A, 4K, 6T
	<p>4(A) The student will use program design problem-solving strategies to create program solutions.</p> <p>4(K) The student will explore common algorithms, including finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average.</p> <p>6(T) The student will identify and use the structured data type of one-dimensional arrays to traverse, search, and modify data.</p>		
	EXAM	1 Day	