Computer Science III and Independent Study
At-A-Glance - Lamar CISD

Ongoing Skills Imbedded All Year	**Professional Standards/Employability Skills/Technical Skills**		
	● Increase and diversify participation in computer science ● Students, regardless of prior experience in computing, will develop confidence using computer science as a tool to express themselves and solve problems, and this confidence will prepare them for success in future endeavors in the field of computer science ● Students will understand the core principles of computing, a field which has and continues to change the world ● Students will be able to develop computational artifacts to solve problems, communicate ideas, and express their own creativity ● Students will be able to collaborate with others to solve problems and develop computational artifacts 1● Students will be able to explain the impact computing has on society, economy, and culture ● Students will be able to analyze existing artifacts, identify and correct errors, and explain how the artifact functions ● Students will be able to explain how data, information, or knowledge is represented for computational use ● Students will be able to explain how abstractions are used in computation and modeling ● Students will learn to be informed and responsible users of technology		
Ongoing Ways to Show	Complete labs and assignment individually or as a team.		
Grading Period	**Unit Name**	**Estimated Time Frame**	**TEKS**
Grading Period 1 29 Days	**Introductory Skills/Set Up**	**4 Days**	7A, 7B, 7C, 7D, 7E, 7F, 7G
	’(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. ’(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. ’(C) The student will create simple programs using a low-level language such as assembly. ’(D) The student will create discovery programs in a high-level language. ’(E) The student will create scripts for an operating system. ’(F) The student will explore industry best practices for secure programming. ’(G) The student will explore emerging industry or technology trends.		
	Independent Learning	**25 Day**	7A, 7B, 7C, 7D, 7E, 7F, 7G
	Students will select 1 topic from the options below and complete a Semester long course and complete a project using the new skills learned. • Unity • Data Science • Mobile App • Web Design- Monet • AI 7(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. 7(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. 7(C) The student will create simple programs using a low-level language such as assembly. 7(D) The student will create discovery programs in a high-level language. 7(E) The student will create scripts for an operating system. 7(F) The student will explore industry best practices for secure programming. 7(G) The student will explore emerging industry or technology trends.		
Grading Period 2 27 Days	**Independent Learning**	**27 Days**	7A, 7B, 7C, 7D, 7E, 7F, 7G
	Students will select 1 topic from the options below and complete a Semester long course and complete a project using the new skills learned. • Unity • Data Science • Mobile App • Web Design- Monet • AI		

	7(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. 7(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. 7(C) The student will create simple programs using a low-level language such as assembly. 7(D) The student will create discovery programs in a high-level language. 7(E) The student will create scripts for an operating system. 7(F) The student will explore industry best practices for secure programming. 7(G) The student will explore emerging industry or technology trends.		
Grading Period 3 **28 Days**	**Independent Learning**	**28 Days**	7A, 7B, 7C, 7D, 7E, 7F, 7G
	Students will select 1 topic from the options below and complete a Semester long course and complete a project using the new skills learned. • Unity • Data Science • Mobile App • Web Design- Monet • AI 7(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. 7(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. 7(C) The student will create simple programs using a low-level language such as assembly. 7(D) The student will create discovery programs in a high-level language. 7(E) The student will create scripts for an operating system. 7(F) The student will explore industry best practices for secure programming. 7(G) The student will explore emerging industry or technology trends.		
Grading Period 4 **31 Days**	**Project Design and Development**	**31 Days**	1A, 1B, 1C, 1D, 1E, 1F, 1G, 1H, 1I, 2A, 2B, 2C, 2D, 2E, 4A, 4B, 4 , 4D, 4E, 4F, 4G, 4H, 5A,5B, 5C, 5E, 5F, 6A, 6B, 6C, 6D, 7A, 7B, 7C, 7D, 7E, 7F, 7G
	Students will complete a project on their own and go through all the phases of Software Development: • Design • Development • Testing • Documentation • Presentation 1(A) The student will identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities. 1(B) The student will examine the role of certifications, resumes, and portfolios in the computer science profession. 1(C) The student will employ effective technical reading and writing skills. 1(D) The student will employ effective verbal and non-verbal communication skills. 1(E) The student will solve problems and think critically. 1(F) The student will demonstrate leadership skills and function effectively as a team member. 1(G) The student will demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science. 1(H) The student will demonstrate planning and time-management skills. 1(I) The student will compare university computer science programs. 2(A) The student will apply object-oriented programming, including data abstraction, encapsulation, inheritance, and polymorphism, to manage the complexity of a project. 2(B) The student will design and implement a class hierarchy. 2(C) The student will read and write class specifications using visual organizers, including Unified Modeling Language. 2(D) The student will identify, describe, evaluate, compare, and implement standard sorting algorithms that perform sorting operations on data structures, including quick sort and heap sort. 2(E) The student will identify and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution. 4(A) The student will identify and use two-dimensional ragged arrays to traverse, search, modify, insert, and delete data. 4(B) The student will describe and demonstrate proper linked list management, including maintaining the head and safe addition and deletion of linked objects. 4(C) The student will create or trace program solutions using a linked-list data structure, including unordered single, ordered single, double, and circular linked.		

	4(D) The student will describe composite data structures, including a linked list of linked lists. 4(E) The student will create or trace program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types. 4(F) The student will create or trace program solutions using sets, including hash and tree-based data structures. 4(G) The student will create or trace program solutions using map style data structures. 4(H) The student will write and modify text file data.5(A) The student will evaluate expressions using bitwise operators. 5(B) The student will evaluate expressions using the ternary operator. 5(C) The student will identify, trace, and appropriately use recursion in programming solutions, including processing binary trees. 5(D) The student will create or trace program solutions using hashing. 5(E) The student will explore common algorithms such as matrix addition and multiplication, fractals, Towers of Hanoi, and magic square. 5(F) The student will create program solutions that exhibit robust behavior by recognizing and avoiding runtime errors and handling anticipated errors. 6(A) The student will use appropriate formatting and write documentation to support code maintenance, including pre- and post-condition statements. 6(B) The student will write program assumptions in the form of assertions. 6(C) The student will write a Boolean expression to test a program assertion. 6(D) The student will construct assertions to make explicit program invariants. 7(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. 7(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. 7(C) The student will create simple programs using a low-level language such as assembly. 7(D) The student will create discovery programs in a high-level language. 7(E) The student will create scripts for an operating system. 7(F) The student will explore industry best practices for secure programming. 7(G) The student will explore emerging industry or technology trends.		
Grading Period 5 **30 Days**	**Project Design and Development**	**30 Days**	1A, 1B, 1C, 1D, 1E, 1F, 1G, 1H, 1I, 2A, 2B, 2C, 2D, 2E, 4A, 4B, 4 , 4D, 4E, 4F, 4G, 4H, 5A,5B, 5C, 5E, 5F, 6A, 6B, 6C, 6D, 7A, 7B, 7C, 7D, 7E, 7F, 7G
	Students will complete a project on their own and go through all the phases of Software Development: • Design • Development • Testing • Documentation • Presentation 1(A) The student will identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities. 1(B) The student will examine the role of certifications, resumes, and portfolios in the computer science profession. 1(C) The student will employ effective technical reading and writing skills. 1(D) The student will employ effective verbal and non-verbal communication skills. 1(E) The student will solve problems and think critically. 1(F) The student will demonstrate leadership skills and function effectively as a team member. 1(G) The student will demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science. 1(H) The student will demonstrate planning and time-management skills. 1(I) The student will compare university computer science programs. 2(A) The student will apply object-oriented programming, including data abstraction, encapsulation, inheritance, and polymorphism, to manage the complexity of a project. 2(B) The student will design and implement a class hierarchy. 2(C) The student will read and write class specifications using visual organizers, including Unified Modeling Language. 2(D) The student will identify, describe, evaluate, compare, and implement standard sorting algorithms that perform sorting operations on data structures, including quick sort and heap sort. 2(E) The student will identify and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution. 4(A) The student will identify and use two-dimensional ragged arrays to traverse, search, modify, insert, and delete data. 4(B) The student will describe and demonstrate proper linked list management, including maintaining the head and safe addition and deletion of linked objects. 4(C) The student will create or trace program solutions using a linked-list data structure, including unordered single, ordered single, double, and circular linked. 4(D) The student will describe composite data structures, including a linked list of linked lists.		

	4(E) The student will create or trace program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types. 4(F) The student will create or trace program solutions using sets, including hash and tree-based data structures. 4(G) The student will create or trace program solutions using map style data structures. 4(H) The student will write and modify text file data. 5(A) The student will evaluate expressions using bitwise operators. 5(B) The student will evaluate expressions using the ternary operator. 5(C) The student will identify, trace, and appropriately use recursion in programming solutions, including processing binary trees. 5(D) The student will create or trace program solutions using hashing. 5(E) The student will explore common algorithms such as matrix addition and multiplication, fractals, Towers of Hanoi, and magic square. 5(F) The student will create program solutions that exhibit robust behavior by recognizing and avoiding runtime errors and handling anticipated errors. 6(A) The student will use appropriate formatting and write documentation to support code maintenance, including pre- and post-condition statements. 6(B) The student will write program assumptions in the form of assertions. 6(C) The student will write a Boolean expression to test a program assertion. 6(D) The student will construct assertions to make explicit program invariants. 7(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. 7(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. 7(C) The student will create simple programs using a low-level language such as assembly. 7(D) The student will create discovery programs in a high-level language. 7(E) The student will create scripts for an operating system. 7(F) The student will explore industry best practices for secure programming. 7(G) The student will explore emerging industry or technology trends		
Grading Period 6 **27 Days**	**Project Design and Development**	**17 Days**	1A, 1B, 1C, 1D, 1E, 1F, 1G, 1H, 1I, 2A, 2B, 2C, 2D, 2E, 4A, 4B, 4 , 4D, 4E, 4F, 4G, 4H, 5A,5B, 5C, 5E, 5F, 6A, 6B, 6C, 6D, 7A, 7B, 7C, 7D, 7E, 7F, 7G
	Students will complete a project on their own and go through all the phases of Software Development: • Design • Development • Testing • Documentation • Presentation 1(A) The student will identify job and internship opportunities and accompanying job duties and tasks and contact one or more companies or organizations to explore career opportunities. 1(B) The student will examine the role of certifications, resumes, and portfolios in the computer science profession. 1(C) The student will employ effective technical reading and writing skills. 1(D) The student will employ effective verbal and non-verbal communication skills. 1(E) The student will solve problems and think critically. 1(F) The student will demonstrate leadership skills and function effectively as a team member. 1(G) The student will demonstrate an understanding of legal and ethical responsibilities in relation to the field of computer science. 1(H) The student will demonstrate planning and time-management skills. 1(I) The student will compare university computer science programs. 2(A) The student will apply object-oriented programming, including data abstraction, encapsulation, inheritance, and polymorphism, to manage the complexity of a project. 2(B) The student will design and implement a class hierarchy. 2(C) The student will read and write class specifications using visual organizers, including Unified Modeling Language. 2(D) The student will identify, describe, evaluate, compare, and implement standard sorting algorithms that perform sorting operations on data structures, including quick sort and heap sort. 2(E) The student will identify and use the appropriate abstract data type, advanced data structure, and supporting algorithms to properly represent the data in a program problem solution. 4(A) The student will identify and use two-dimensional ragged arrays to traverse, search, modify, insert, and delete data. 4(B) The student will describe and demonstrate proper linked list management, including maintaining the head and safe addition and deletion of linked objects. 4(C) The student will create or trace program solutions using a linked-list data structure, including unordered single, ordered single, double, and circular linked. 4(D) The student will describe composite data structures, including a linked list of linked lists. 4(E) The student will create or trace program solutions using stacks, queues, trees, heaps, priority queues, graph theory, and enumerated data types.		

	4(F) The student will create or trace program solutions using sets, including hash and tree-based data structures. 4(G) The student will create or trace program solutions using map style data structures. 4(H) The student will write and modify text file data. 5(A) The student will evaluate expressions using bitwise operators. 5(B) The student will evaluate expressions using the ternary operator. 5(C) The student will identify, trace, and appropriately use recursion in programming solutions, including processing binary trees. 5(D) The student will create or trace program solutions using hashing. 5(E) The student will explore common algorithms such as matrix addition and multiplication, fractals, Towers of Hanoi, and magic square. 5(F) The student will create program solutions that exhibit robust behavior by recognizing and avoiding runtime errors and handling anticipated errors. 6(A) The student will use appropriate formatting and write documentation to support code maintenance, including pre- and post-condition statements. 6(B) The student will write program assumptions in the form of assertions. 6(C) The student will write a Boolean expression to test a program assertion. 6(D) The student will construct assertions to make explicit program invariants. 7(A) The student will analyze and create computer program workflow charts and basic system diagrams, documenting system functions, features, and operations. 7(B) The student will gather requirements, design, and implement a process by which programs can interact with each other such as using interfaces. 7(C) The student will create simple programs using a low-level language such as assembly. 7(D) The student will create discovery programs in a high-level language. 7(E) The student will create scripts for an operating system. 7(F) The student will explore industry best practices for secure programming. 7(G) The student will explore emerging industry or technology trends		
	Project Presentations	**10 Days**	3A, 3B
	Students will complete a project on their own and go through all the phases of Software Development: • Design • Development • Testing • Documentation • Presentation 3(A) The student will use networked tools for file management and collaboration. 3(B) The student will work in software design teams.		