Introduction to Computer Science

Course Description	
This course introduces students to the foundations of computer science with a focus on how computing powers the world. Students will explore computer science terminology and concepts and apply them to a culminating project using programming language to solve a problem.	
Course Code:	**8675309**
Standard 1	
Develop foundational skills and knowledge in computer science.	
Indicator 1.1	Map a brief overview of computer science history.
Indicator 1.2	Show how binary bits can be used to create numbers, characters, letters, images, audio and video.
Indicator 1.3	Consider the pros and cons of the variety of computing languages.
Indicator 1.4	Appraise the current opportunities available in computer science career fields.
Indicator 1.5	Articulate common vocabulary in computer science. (abstraction, decompose, patterns, algorithms, byte, bit, binary, computational thinking, compression, etc.)
Indicator 1.6	Learn the steps of a design cycle and how it works and how it works as a practical problem solving method.
Indicator 1.7	Understand what comprises a computer: input device, memory, output, control unit, arithmetic/logic unit. (Von Neumann architecture.
Indicator 1.8	Explore the impact that modern computing has on the world.
Standard 2	
Design, use and evaluate abstractions that model real-world problems and use the models to make predictions about the world.	
Indicator 2.1	Define abstraction as it is used in computer science. (Examples: word processing represents paper, disease spread simulation, video game is a world)
Indicator 2.2	Identify abstractions.
Indicator 2.3	Create an abstraction to model something in the real world.
Indicator 2.4	Use the abstraction to draw conclusions about the real world.
	https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1
Standard 3	
Use computational thinking to makes sense of knowledge to accomplish a goal or task or solve a problem.	
Indicator 3.1	Define the four steps of computational thinking. (Decompose, Patterns, Abstraction, Algorithm)
Indicator 3.2	Use the steps of computational thinking to solve a problem.
Standard 4	
Construct a computational artifact using a programming language. (computational artifact is the final product of what is programmed)	
Indicator 4.1	Develop language literacy for a programming language. (debugging, variables, sequence, input-output, conditionals, loops, syntax, functions, algorithms)
Indicator 4.2	Use computational thinking to explain how simple algorithms work and to detect and correct errors in algorithms and programs.
Indicator 4.3	Design, write and debug simple programs that accomplish specific goals.
Indicator 4.4	Employ pair programming to design, write and debug a program that accomplishes a specific goal. (Driver/Navigator Description)
Indicator 4.5	Using a programming language, design and develop an independent program to solve a problem.
Contributors:	Resources: code.org, Snap, Scratch, App Inventor, javascript, Python
Susan Schiltz	
Tamara Zentic	
Ramsey Young	