Computational Thinking in STEM

Skills Taxonomy

Table of Contents

1. Data and Information Skills

Data play a critical role in the STEM fields. There are many skills associated with collecting and using data effectively and efficiently. Many of these skills overlap with Computational Thinking skills; together, Computational Thinking skills and data skills will empower learners to ask and answer challenging STEM questions. Computational Thinking skills are used in all facets of data-related STEM work from the initial data collection phase all the way through drawing conclusions and sharing your findings.

Keywords: Data; Data Collection; Data Analysis; Data Visualization;

1a. Collecting Data
Data can be collected through observation and measurement. Different types of STEM inquiries require different data collection procedures and result in various forms of data. Computational tools are useful in all the different phases of data collection, from the design of the collection procedures through the execution, recording and storage of the resulting data.
Students will understand that… • Collecting data includes questions of accuracy, precision, validity, and data storage. • Data can take different forms including numbers, text, images, and audio or video formats. Each of these forms comes with a different set of computational tools for collection and storage. • Computational devices can assist in data collection. They are especially useful when collecting large data sets and when a high degree of precision is desired. By precision we are referring to the specificity of the resulting measurement. • Computers can automate the process of data collection to increase the efficiency and validity of the collected data. By validity we are referring to the trustworthiness of the data – whether the data represent what we believe they do. • Using a computational device to conduct data collection procedures can increase the accuracy of the measurement as well as remove human error from the collection process. By accuracy we are referring to the closeness of the measurement to the true value of what is being measured. • Underlying randomness can produce different results from the same initial configuration. Thus, it is important to conduct data collection procedures multiple times to identify potential sources of error inherent in the data collection process. • The validity and accuracy of data can be affected by the procedures followed in the data collection process. When using a computational device to assist in these processes, the problem can be magnified due to repetition, thus it is important to regularly review the data that have been collected and the procedure being followed to ensure the data collection process is proceeding as expected. • Data must be stored for future use and analysis; digitally storing data on computers is a safe, efficient, and reliable approach that facilitates future access.

Students will be able to… • Collect data using a variety of computational tools, e.g. digital sensors, computer simulations or models, and spreadsheets. • Decompose a large data collection strategy into a systematic set of sub-tasks that can be carried out to achieve the larger data-collection goal. • Propose a data collection procedure (step-by-step instructions) that could be followed (by either a human or machine) to gather data to investigate a specific scientific question. • Describe what types of data collection activities are best done using a computer, and what types of data collection tasks require, or are more easily accomplished by a human (e.g. classification of visual characteristics vs. specific repeated steps). • Carry out a specified data collection task multiple times, record data, and explain potential sources of variation and error in the resulting measurements. • Assess and critique data collection practices with respect to validity, precision and accuracy. Students will also be able to explain how they could assess a data collection procedure with respect to these three dimensions. • Define an automated procedure that a computer could carry out to gather a particular set of data or produce a data set that matches a set of pre-defined criteria. • Store collected data in an appropriate form on a computational tool.

1b. Generating Data
Sometimes STEM researchers are investigating phenomena that cannot be easily observed or measured. For example, to investigate how galaxies form, we must use computational models to generate data because the phenomenon takes millions of years to occur. Computational tools make it possible to create data sets based on generative descriptions of data, making it possible to investigate such phenomena.
Students will understand that… • Computers are powerful tools for creating data sets given a generative data description that can be executed. Computers often rely on iterative logic to create data sets. • Randomness plays an important role in creating realistic data sets and must be explicitly included in computationally executable generative descriptions of data. • When writing a program to create a data set, you can control all aspects of the data set including its size, distribution and the range of values it includes. Moreover, you can define what form the data is output in and how/where it will be stored. • When using a computational tool to generate data, the output of the procedure will be the data you wish to investigate, so questions of format and storage must consider future uses of the data.
Students will be able to… • Assess and critique data creation practices with respect to the desired characteristics of the data set needed to investigate the initial question. • Define an automated procedure that a computer could carry out to gather a particular set of data or produce a data set that matches a set of pre-defined

criteria. • Produce an algorithm to follow when trying to create a computer-based generative representation for a data set that matches data collected in the real world. • Introduce randomness into a computational generative description of data and discuss the role it plays in the data generation process. • Discuss the role iterative logic plays in generating a set of data using a computer

1c. Choosing Efficient Data Structures
Data structures provide different ways to store and organize data on a computer. Using apt data structures is a key to designing efficient algorithms, especially when working with large datasets. STEM fields often deal with massive amounts of data, so being able to select, or if necessary create, efficient data structures is critical.
Students will understand that… • Data structures are computation entities (or collections of entities) used to store, manipulate, search and access data in a program. • Data structures are implemented using algorithms. Different algorithms have different efficiencies for the amount of time and memory that it takes in terms of the size of input data. • Different data structures have different strengths and weaknesses. Some are good for iterating through, others are good for inserting and deleting elements quickly, while others are good for searching efficiently. • There is a set of conventional and commonly used data structures (ex. lists, queues, stacks, sets, trees, maps), but it is sometimes necessary to develop custom data structures. Custom data structures can be defined to reflect relationships that exist within a data set or can be customized to perform well for the specific task at hand.
Students will be able to… • Identify different types of data structures that are commonly used, such as arrays, linked lists, stacks, queues, trees, and maps. • Suggest appropriate data structures for a certain problem or task depending on what the data will be used for. • Use data structures in a computer program or a pseudo-code and write procedures for accessing and manipulating the data in the structure. • Translate a set of data from one type of data structure to another. • Propose a custom data structure for a specified task and justify decisions made in creating the new data structure.

1d. Manipulating Data
Data are not static; they are malleable. In STEM fields it is important to be able to manipulate data so that it is more useful and more insights can be gained from data. Computational tools make it possible to efficiently and reliably manipulate large and complex sets of data.
Students will understand that… • By rearranging or reorganizing data new insights can be found. Computational tools can assist in the data manipulation process.

- Data often have to be manipulated and organized after being collected (or generated) and prior to it being analyzed. "Raw" data are not always ready for analysis. Data must be "cleaned" before they can be used.
- There are systematic strategies that can be employed to make data more useful and more usable, strategies include:
  - Standardizing data formats
  - Normalizing data points and removing duplicates
  - Logically sorting and grouping data points
  - Identifying outliers that may be the result of collection or measurement errors.
- Computational tools (in particular spreadsheet and database programs) can be very useful for manipulating data, especially when working with large, complex data sets.
- Data Manipulation and Data Analysis inform and influence each other, and often co-occur.
- Many computer programs used in data analysis require data to be in a very particular format; this requires the initial data to be manipulated so that these tools can be used.
- Computers can help improve the accuracy and precision of a data manipulation task, especially when working with a large data set.
- During the data manipulation process, information should be preserved. That is to say, no information should be lost, before and after the transformations.

Students will be able to…

- Use computational tools (such as Microsoft Excel) to manipulate a set of data.
- Transform a dataset to match a specified format so it can be used as input into a specified analysis tool.
- Sort datasets based on different criteria.
- Normalize a set of data, including standardizing names, forms and removing duplicates.
- Locate data anomalies and provide a justification for leaving the outlying data points in the data set or removing them based on the intended use of the data.
- Manipulate a dataset as a whole rather than each of the data points.
- Discuss the pros and cons of different computational tools using for cleaning the data.

1e. Analyzing Data

The true power of data lies in the information that can be gleaned from them through analysis. There are many strategies that can be employed when analyzing data for use in a STEM project. Computers can help analyzing data in a more reliable, effective manner.

Students will understand that…

- Data analysis is a prerequisite for drawing conclusions or making hypothesis.
- During the data analysis process, individual data points might shift, change form, or otherwise be altered, but the underlying information should not change.
- Computational tools can help us with analyzing large amounts of data.
- Data analysis often includes looking for patterns and underlying structures in a data set.

- To study a data set, it can be described in many forms including mathematical equations, spreadsheet formulas, computer programs, or as collections of individual data points. One potential output of data analysis is a generative description of that data that can be used to create similar data sets.
- One dataset can be interpreted and analyzed in many different ways, depending on the question that is being investigated.
- Conclusions drawn from data analysis are only as reliable as the data from which they were drawn.
- Many computational tools offer functionality to assist in analysis of data, including tools for graphing, charting, filtering and conditionally displaying data.
- Creating visualizations of data can be a useful strategy when conducting analysis.
- Data analysis and data manipulation often proceed hand-in-hand. Manipulation helps analysis, analysis informs manipulations.

Students will be able to…

- Suggest and implement a successful data analysis strategy for the question that is being investigated.
- Select and use an appropriate dataset to be analyzed for a specific problem.
- Search through a dataset to find patterns and anomalies in the data using computational tools.
- Leverage computational tools to assist in data analysis.
- Produce a data visualization and discussion how the chosen visualization can assist them in conducting their data analysis.
- Save the original dataset and new altered data sets in a way that necessary information is always retrievable.

1f. Visualizing Data
Data can take many forms and can be displayed many ways. In STEM fields, creating visualizations is a powerful strategy for both analyzing and sharing data. Computers are powerful tools for designing and implementing both static and dynamic data visualizations.
Students will understand that… • Data visualizations can be used to highlight interesting or important characteristics of data that can be useful for analysis or for presenting your data to others. • Computers can be used to create many different types of data visualizations including graphs, charts, plots, diagrams, movies and figures. • Computers make it possible to generate interactive and dynamic data visualizations that can be an effective and engaging way to explore a set of data. • Interactive visualizations can be rendered in one, two, or three dimensions. When studying a phenomenon that occurs in nature, it is often powerful to render it in 3-dimensions as it creates a more realistic representation of how the phenomenon actually occurs. • When using a computer to visualize a set of data, you need to give the computer explicit instructions for what data to include and how it should be displayed. There are many characteristics that can be manipulated to highlight certain aspects of the data; including size, color, shape, position, and text labels.

• When creating a visualization, you often only display a subset of the entire data set. Which data and how much of it you include depends on the information you are trying to convey and the type of visualization you are creating.
Students will be able to… • Use computational tools to design and implement data visualizations that highlight interesting aspects of a provided data set. This includes an accompanying write-up that articulates what the visualization is showing. • Define a set of parameters so that only a subset of a larger data set is included in a visualization. These parameters might require using conditional logic, developing bucketing strategies, or defining upper and lower bounds for inclusion. • Articulate the process followed in creating a visualization so that others can create similar products. • Specify conditions to have data points display differently based on specific characteristics of the data. • Assess the strengths and weaknesses of different visualizations for the same set of data.

2. Problem Solving Skills

Problem solving is integral to STEM practice. The process of researching and developing understandings of STEM phenomena is full of problems and challenges that must be overcome. Computational tools provide STEM practitioners a powerful leverage in the pursuit of knowledge and understanding, but the challenge remains of how best to employ these tools to solve the problems at hand. Thus, an important set of skills associated with Computational Thinking in STEM include computational approaches to problem solving skills to solve the challenging problems faced by STEM practitioners as they conduct their work.

Keywords: Problem Solving; Troubleshooting; Debugging, Problem Decomposition, Algorithms; Problem Simplification

2a. Decomposing Problems into Subproblems / Developing Modular Solutions
In STEM, we are often concerned with problems that are too complex to be solved all at once. Problem decomposition (or developing modular solutions) is the ability to break down a problem into smaller pieces that can be more easily solved. Developing computational solutions in a modular way makes it possible to develop the solution incrementally, test the components independently, and more easily reuse portions of the solution in future problems.

Students will understand that…
• An effective, and sometimes necessary, approach to solve a complex problem is to break it down into smaller problems.
• Decomposing a problem into smaller pieces allows the large problems to be solved faster, as the subproblems can be solved in parallel.
• Decomposition can be iterative; sometimes subproblems are still too large to be solved and thus, need to be further decomposed.
• Decomposing problems into subproblems enables multiple team members to work on different components of a problem simultaneously.
• Encapsulating behaviors of a program into individual modules is an effective strategy for building large, complex computer programs.
  o Modularity helps make programs more flexible. Having small modular pieces allows them to be reused, reorganized and adapted for future uses.
  o Modular solutions are easier to test for correctness and fix than large, monolithic programs because modular solutions make it possible to isolate and address bugs.

Students will be able to… • Decompose a problem into a set of subproblems that are easier to solve and result in a global solution for the initial problem. • Use the solutions from subproblems to answer, or make progress, on the initial, larger problem. • Identify the dependencies between subproblems and determine the pieces that can be solved in parallel. • Break down a problem into tasks in such a way that the workload can be distributed over members of a group.

- Design and implement modular solutions in such a way that they can be reused in future solutions.
- Discuss how each module of a solution works individually and how the modules work together to generate the global solution to the problem.
- Explain the advantages of modularly designed computer programs.

2b. Reframing Problems into Known/Familiar Problems
Reframing a problem into a form where the solution is already known/understood is a broadly applicable problem solving approach. Repurposing solutions from similar problems is especially effective when generating computational solutions for new problems as computational tools can often be reused with little or no alteration. This is a common practice in the STEM fields for the investigation of new phenomena or solving problems that are similar to other, already solved, challenges.
Students will understand that… • There are frequently commonalities across problems, even if they appear different at first glance. Once the commonalities are recognized, they can be leveraged to apply approaches/solutions from a solved problem to the unknown problem. • Similarities in the underlying principles or concepts involved in the solution to the problem are more important than surface similarities when deciding if a known solution will be applicable in a novel context.
Students will be able to… • Discuss strategies that can be used to identify commonalities across problems. • Identify and then background unimportant features of related problems when searching for common approaches to solve the problems. • Solve a problem by reframing it into a similar problem that is already understood.

2c. Simplifying Complex Problems
A useful initial step in solving challenging problems is to identify the source of complexity so that it can be simplified. By knowing the source of complexity (i.e. what makes the problem hard), one can simplify the problem by isolating the complex details, ultimately making the problem easier to solve. This is an important skill in STEM, as many STEM problems are complex by nature and it is very difficult or impossible to solve them without teasing out the unnecessary details.
Students will understand that… • Problems are sometimes less complex than they seem; identifying the parts of the problem that are relevant and focusing on those can make a large, complex problem simpler. • One effective strategy to solve a complex problem is to first find a solution for a simplified version of the problem by temporarily ignoring some details of the problem at hand. Upon solving the simplified problem, you can then reintroduce the aspects of the problem you had initially ignored.
Students will be able to… • Identify the source(s) of complexity in a problem. • Recognize what makes solving a problem easier or harder, such as the amount of detail provided, the number of unknowns, the parts that can be automated, and existence of software designed to help solve the problem.

- Determine if and how the most complex parts of a problem can be circumvented to solve a simplified version of the problem that yields insights into the initial problem.

2d. Generating Algorithmic Solutions
An algorithm is a set of instructions that defines a step-by-step approach to solving a problem. Generating algorithms is an important problem solving strategy, especially when using a computational device to automate part of, or execute the entirety of the proposed solution. Additionally, algorithmic solutions are important when the approach will be repeated in the future either by you or by others. In STEM fields, many problems require algorithmic solutions, and many solutions are used by others, making a clearly defined algorithm essential.
Students will understand that… • Algorithm design involves outlining the steps that should be followed to solve a problem or accomplish a task. • Algorithms can include conditional logic, recursion, iteration, and other principles from computer science. • A successful algorithm is the one that is clear and unambiguous at each step and achieves the desired result. • Writing pseudo-code is an effective way to outline an algorithm before implementing it on a computer or communicating it to team members. • The specificity of an algorithm depends on who or what is going to carry the algorithm out. Computers require very explicit steps. • Algorithms can be expressed in a number of different ways including pseudo-code, executable computer code and flow charts.
Students will be able to… • Propose an algorithmic solution to a problem. • Leverage constructs such as conditional logic and iteration when defining an algorithm. • Express algorithmic solutions using representational systems that can be executed by a computer. • Leverage visual representations, like flowcharts, to define and share a proposed algorithmic solution.

2e. Applying Algorithms
In solving a problem, one can apply different algorithms, each of which may have its own set of strengths and weaknesses. In STEM fields, there are many commonly used algorithms and being able to understand, carry out, and assess these algorithms is an important skill.
Students will understand that… • An algorithm that is designed to solve a certain problem can also be applied to other problems that have similar patterns. • Algorithms can take many forms, including computer code, instructions written in prose, and graphical flow charts. • While more than one algorithm may produce the desired result, there are other criteria than can be used to assess an algorithm, including efficiency, ease of

implementation, and cost of implementation (in terms of time, money, and other resources).
- Often it is possible to solve a problem by modifying an existing algorithm, as opposed to creating a new algorithm from scratch.

Students will be able to…
- Read through and understand algorithmic solutions generated by others.
- Modify a given algorithm to accomplish a different, but related goal.
- Assess the strengths and weaknesses of different algorithms that can be used to solve the same problem.

2f. Assessing Different Approaches/Solutions to a Problem

When there are multiple approaches to or solutions for a problem, it is important to be able to assess the options and make an informed decision about which route to follow. This is important in STEM fields, where there is often more than one way to solve a problem. This is especially true in generating computational solutions, as there are often many software alternatives available. Thus, students should be able to assess different approaches based on their requirements, the constraints of the problem, and the available resources.

Students will understand that…
- Solving a problem is not only about finding "a" solution but finding "the best possible" solution for that problem.
- There are often multiple approaches and methods that can be used to solve a problem; depending on the constraints of the problem (e.g., accuracy, precision) and resources available (e.g., CT tools available, time, cost, ease of implementation), some approaches will be more ideal than others.

Students will be able to…
- Propose and consider the trade-offs between different approaches and solutions to a problem.
- Identify criteria that are important to consider when deciding which approach/solution to adopt, based on the conditions of the problem and the resources available.
- Justify the approach that is chosen with respect to the constraints of the problem and the resources available.

2g. Choosing Effective Computational Tools

In STEM fields, the solution to a problem can be found using a number of different approaches and a number of different computational tools. Therefore, it is important to be able to assess the strengths and weaknesses of different computational tools, so you can find the appropriate tool for a specific problem.

Students will understand that…
- Computational tools include a wide variety of computer technologies, including: programming languages, software, hardware, or a combination of the above.
- Computational tools have unique sets of features and capabilities as well as limitations and constraints. These differences make certain computational tools better suited for solving certain types of problems than others.
- There is often more than one computational tool that can be used to solve a

particular problem.

- It is sometimes necessary to use multiple computational tools to solve a single problem.
- There are many dimensions along which one can assess a computational tool including: speed, power, cost, ease of use, the input/output formats supported, etc.

Students will be able to…

- Propose candidate computational tools for a specific problem.
- Discuss advantages and disadvantages of a particular computational tool.
- Map a certain problem to a set of tools that could be used to solve that problem.
- Recognize the commonalities and differences between a set of computational tools.

2h. Troubleshooting/Debugging

Troubleshooting is an important problem solving skill, which broadly captures the process of figuring out why something is not working/behaving as expected. There are a number of strategies one can employ while troubleshooting a problem, including clearly identifying the issue and systematically testing the system to isolate the source of the error. In computer science this skill is often referred to as "debugging". In STEM fields, the ability to troubleshoot a problem is important as unexpected results and incorrect behavior are frequently encountered.

Students will understand that…

- Troubleshooting is an important component of designing, executing and interpreting results from a STEM inquiry when unexpected challenges or behaviors emerge.
- An important step in the process of designing a new system or product is testing the design and troubleshooting discovered errors.
- When troubleshooting a problem, it is important to follow a systematic approach for isolating the source of the error.
- It is often useful to break down a complex process into smaller sub-processes and check the state of system properties (or variables) at these sub-intervals. This is especially true when testing an experimental design or debugging a computer program.
- Troubleshooting is an iterative process that includes a number of steps including: diagnosing the problem; fixing the errors; and testing the implemented solution.
- Getting an experimental design to lead to a solution or a computer program to run to completion is only the first step. You must also test your experimental set-up or program to ensure it produces reasonable outcomes.

Students will be able to…

- Systematically determine the source of an error or malfunction in a system.
- Isolate different parts of a system to assess each part's behavior individually.
- Propose different approaches to identifying errors and testing potential solutions.
- Perform an iterative process of fixing the errors in an algorithm, experimental design, or a computer program.

3. Computational Modeling Skills

The use of computational models is a central practice in the investigation of STEM phenomena. We use the term computational models to refer to non-static computer-based simulations or models of real world phenomena. These tools make it possible to investigate questions and test hypotheses that would otherwise be too expensive, too dangerous, too difficult or entirely not possible to carry out otherwise. Computational tools make the practice of modeling and simulation possible on a scale that was not possible before. However, one must keep in mind that computational models are not the real world and therefore be aware of how a model or simulation differs from the real world. Computational Thinking skills play a large role in unleashing the power of computation for the investigation of STEM phenomena through computational models.

Keywords: Computer-based Models; Computational Simulations; Model-driven Inquiry; Model Evaluation; Model Construction;

<table>
<tr><th>3a. Using Computational Models to Understand a Concept</th></tr>
<tr><td>Computational models are tools that can be used to deepen understanding of STEM concepts. Such tools make it possible to recreate STEM phenomena in environments that support systematic and careful investigation that give the user far more control than would be possible by studying the phenomena in the natural world.</td></tr>
<tr><td>Students will understand that…
• Computational models are powerful tools for making sense of the world and that computational tools make this practice possible.
• Computational models make it possible to systematically manipulate variables to see how different components impact the larger system.
• Computational models of real world phenomena often make it possible to investigate and explore in ways not possible in the real world. Barriers computational models can overcome include:
    o Physical limitations, phenomena too big or small to study naturally
    o Temporal constraints, phenomena that are too fast or too slow to be studied naturally
    o Cost limitations, phenomena too costly to systematically investigate</td></tr>
<tr><td>Students will be able to…
• Use a computational model to gain an understanding of the phenomena or concept the model simulates.
• Understand and explain the role different inputs into a computational model play in its overall behavior and functioning.
• Explain the relationship between different modalities and representations provided by the model's interface.
• Interpret results of a given run of a simulation in context of concept being investigated.</td></tr>
</table>

3b. Understanding How and Why a Computational Model Works
It is important to understand how and why a model works to be able to determine what conclusions can be drawn from it and to what extent the findings from the model can be applied. Developing this type of understanding is valuable in STEM fields as it can give a deeper insight into the nature of the phenomena and the power of the model.
Students will understand that… • Computational models are not the real world; therefore, to make a better sense of the model, you must be aware of how it differs from the real world. • Computational models usually simplify the real world phenomena by highlighting the key features and backgrounding the unnecessary details. • The observed behavior of a computational model is a result of the collective underlying rules and behaviors. This information either comes from accompanying descriptive text to the model, from studying the actual code, or through exploring and using the model itself. • Although the computational models generally simulate complex behaviors, the underlying rules can sometimes be quite simple. • By understanding the rules underlying the observed behavior of a model, you are better able to interpret and apply insights gleaned from the model.
Students will be able to… • Identify and discuss the underlying rules of a computational model as well as the roles they play in the model. This can result from studying accompanying documentation or the code that drives the model. • Explain how different sets of configurations of the parameters a model influence its resulting behaviors. • Identify what states a model can be in and discuss what causes the model (or an element of that model) to change from one state to another during a single run of the model. • Make sense of multiple runs of a simulation and discuss the cause or lack of variation in the resulting output. When randomness is present in the model, students will be able to identify the source of this randomness and the role it plays in the model.

3c. Assessing Computational Models
A key skill in using computational models is to understand how a given model relates to the phenomenon being modeled. Central to this is the ability to validate, verify and calibrate a model as well as identify strengths and weaknesses of a given model. This is especially true in STEM fields as models often are designed to recreate phenomena from the natural world.
Students will understand that… • Models are not exact replicas of the phenomena and are often simplified versions of the actual phenomena. • Modeling tools have strengths and weaknesses, making some modeling tools better suited for modeling some phenomena than others.

- Sometimes models foreground some aspects of a phenomenon while background, or altogether ignoring, other parts of the phenomenon.
- Empirical data is a key resource when assessing a model. When using a computational model, it is important to verify the model by comparing it to the real world scenario.
- The uncertainties on inputted values contribute to the uncertainty on outputted values in a model; i.e., large uncertainties on inputted values result in a lower confidence in the accuracy of the outputted results.

Students will be able to…

- Identify where models are accurate and where they differ from the phenomena they are meant to demonstrate.
- Assess a computational model with respect to validity, accuracy, and precision.
- Understand and articulate the implications of shortcomings of models.
- Verify a model against a dataset produced through observations of the phenomena being modeled.
- Discuss what types of research questions a model can address based on the controls the model provides and the data it produces.
- Choose the best available modeling package to simulate a phenomenon by comparing the capabilities and constraints of each package.
- Articulate the types of STEM questions that can be investigated by a particular computational model.

3d. Using Computational Models to Find and Test Solutions

Computational models are tools that can be used to test hypotheses and discover solutions to problems. They make it possible to try many different solutions quickly, easily and inexpensively. This is an important technique for investigating problems in STEM fields.

Students will understand that…

- Computational models are powerful exploratory tools for finding and testing solutions to STEM problems.
- Computational models are often an easier, cheaper and faster way to test potential solutions than actually attempting the solution in the real world.
- There may be more than one solution to a problem; computational models can help identify the strengths and weaknesses of different proposed solutions.
- Many alternatives should be tested before settling on a solution. This includes:
  - Exploring parameter/problem space
  - Adding/removing variables
  - Considering various strategies/approaches to the problem at hand
- There are many types of modeling packages that differ in a variety of ways:
  - The paradigm and its underlying mechanisms (e.g., agent-based modeling vs. mathematical modeling vs. simulations)
  - The technology used for implementation
  - Granularity of entities (different levels; mesh size)
  - Fidelity to the real world phenomena

Students will be able to…

- Use a computational model to find and justify a solution (or many solutions) to a STEM problem.

• Interpret and discuss the implications of solution found using models. This includes making a decision about which of a number of solutions to adopt to the particular problem under investigation. • Discuss the advantages of using computational models to find solutions to problems (being cheap, easily reproducible, scalable, and controllable).

3e. Building New and Extending Existing Computational Models
While many pre-existing models can be found, it is often necessary to create a new model to meet the specific requirements of the phenomena or problem you are investigating. Other times, when a useful simulation or modeling package does exists, you might need to manipulate the underlying rules and extend the model to be able to explore a specific question. The ability to build new models (or extend existing models) allows STEM professionals to independently explore new phenomena and test out new potential solutions.

Students will understand that…
• In developing new models and extending existing ones, the level of simplification you use influences the extent of your findings and the nature of conclusions you can draw from your model.
• Computational models can often be extended by:
    o Introducing new rules or objects to the model
    o Adding specificity to existing rules/objects in the model
    o Adding features to support new scenarios
• When creating a new model, sometimes the conceptual model is fully understood before implementation begins (top-down design), other times the conceptual model and the coded model co-evolve (bottom-up design).
• The different CS and problem solving techniques discussed in sections "computer skills" and "problem solving skills" can be employed to create, modify, and extend computational models

Students will be able to… • Articulate different ways an existing model can be extended and what the resulting changes will add to the model with respect to types of questions that can be answers and concepts that can be investigated. • Develop a step-by-step plan for creating a new model. • Determine what details of a problem or phenomenon need to be foregrounded and which ones should be backgrounded to investigate or answer a specific STEM question. • Identify what controls need to be available to the user to investigate a particular phenomenon. • Conceptualize and communicate the design of a model before starting to implement it.

4. Systemic Thinking Skills

STEM phenomena rarely involve individual objects acting in isolation; instead they are the result of the interactions between interrelated elements, which constitute a system. These elements could be anything from organisms in an ecosystem to mechanical components of a car engine to chemical elements in a solution. Across these varied situations, there is a set of Computational Thinking skills that are useful for identifying different elements of a system and understanding how they function and interact.

Keywords: Complex Systems; Inputs/Outputs; Emergence; Micro-Macro Interactions

4a. Understanding the Relationships within a System
When investigating phenomena that are the result of many elements interacting within a system, it is important to be able to identify the different elements/parts of the system and articulate how they work together. Computational tools are useful for developing an understanding of such systems and investigating the role that different elements play within the system. In STEM fields, this skill is critical as many of the phenomena under investigation are composed of many different elements acting in concert.
Students will understand that… • Systems can be composed of a variety of elements that can play similar or different roles. • Sometimes different types of elements play complimentary roles, helping each other; while other times elements can play adversarial roles and compete within a system. • A change to one element of the system can result in a change into the output of the system or possibly to other elements of the system. • Sometimes, a small change in a system can result in a big change in the output or other parts of the system. • Computational tools (such as simulations of phenomena) can provide controlled environments with which to study and investigate the different elements of a system and the roles that each play. • The characteristics and properties of a system (as a whole entity) are different from those of the sum of the interrelated elements of the system.
Students will be able to… • Identify constituent elements of a given system. • Describe how different components of a system interact with each other. • Propose approaches one can take to identify what role a given element plays within a system. • Predict how a change in one part of the system will propagate and what effects it will have on other parts of the system.

4b. Thinking in Levels
Systems (like ecosystems, chemical solutions, and mechanical devices) can be viewed at different levels, from a micro-view that looks at the smallest possible element of the

system, all the way up to a macro view that looks at the system as a whole. Different levels afford different insights and have both advantages and disadvantages. In STEM fields, it is important to be able to negotiate these levels and use these different lenses to your advantage in developing an understanding of the phenomenon at hand.
Students will understand that… • Different views of a system have different strengths and weaknesses. Some information about a system only makes sense from a macro (or aggregate) view, while others are unique to a finer, more micro view. • An effective strategy for understanding a system or solving a problem is to view them from multiple levels as different levels may yield different insights.
Students will be able to… • Propose ways of measuring aspects of the system at different levels. • Identify different levels within a system and discuss their properties/qualities. • Associate characteristics/measures of a system with the appropriate level. • Discuss how different levels of a system are related. • Switch between different levels when viewing a system to solve a problem.

4c. Investigating a Problem/Phenomenon by Viewing It as a System
In the STEM fields, many phenomena are the result of systemic interactions. One technique for investigating a phenomenon or solving a complex problem is to design a simplified, representational system. These investigations rely on the ability to define and measure inputs to and outputs from a system, as well as skills involved in isolating different components of the system. Computational tools are especially useful in such investigations as they are useful for automating investigations, taking precise measurements, and modeling systems for further investigations and hypothesis testing.
Students will understand that… • One approach in solving a problem (or investigating a phenomenon) that involves many interactions is to view it as a system. Investigating a system is often easier than investigating a problem by itself; you can make sense of a system by focusing on its elements and the way that these elements work together. • Systems have inputs and outputs. When investigating a system, varying the inputs and measuring changes in output is one mechanism for developing an understanding for how the system works. Computational tools are especially useful for systematically testing combinations and variations of system inputs. • Systems can take more than one input and produce more than one output. • Systems can be composed of small self-contained subsystems that operate within the larger system. These systems can be investigated independently of the larger system, with the findings being used to understand the larger system as a whole.
Students will be able to… • Transform a complex problem into a systemic representation to solve it. • Identify the details of a problem or phenomena that can be transformed as system elements and determine the relationships between these elements. • Identify the full set of inputs and outputs of a given system. • Design a series of tests to measure different outputs from a single system and how they relate to the inputs to that system. • Isolate different subsystems of a larger system and test them individually.

5. Concepts from Computer Science

One important aspect of computational thinking in STEM is the ability to effectively take advantage of computational power by employing tools and strategies from the field of computer science. This knowledge is useful for developing a deep understanding of STEM phenomena and pursuing new and interesting STEM questions. Thus, STEM practitioners benefit from developing computer science skills that enable them to express ideas in a form that a computer can interpret and execute to investigate STEM phenomena.

Keywords: Computer Science; Computational Tools; Programming; Conditional Statements; Iterative Logic; Abstraction

5a. Interpreting Instructions Written for a Computer
Computer programs are frequently shared within a community so others can make use of them. Therefore, it is important to be able to understand and interpret instructions written for a computer in order to learn from and potentially use or extend them. This is especially true when investigating a STEM phenomenon with the assistance of a computational device.
Students will understand that… • There are a number of strategies for comprehending a computer program including: reading the code; reading in-line comments in the code; reading accompanying documentation; searching the Internet for support forums. • While different languages have their own syntax and primitives, there is a set of common features that exist across many languages. • Before using a set of instructions written for a computer in your own work, you should first develop a general understanding of exactly what the program does. • Before modifying or extending a program it is necessary to have a thorough understand of both what it does and exactly how it does it.
Students will be able to… • Comprehend a computer program written by another person and explain what the program does and how it does it. • Step through a computer program in a way that follows the order of execution. • Read through a computer program and explain what the program does.

5b. Defining Instructions for a Computer
Computers require unambiguous instructions in a specific syntax; for a computer to carry out commands, you must represent your problem using a language or a representational system that the computer understands. Defining instructions for a computer is central for taking advantage of computational power and is important in STEM fields.
Students will understand that… • Instructions given to a computer must be expressed using a language or representational system the computer can understand. • Computers require specificity; instructions for a computer must be unambiguous.

- There exist a set of programming constructs that are common to many languages, such as conditional blocks, iterative loops, variables, and functions.
- There are a wide variety of computer languages; each language has its strengths and weaknesses. Not all computers understand the same languages.

Students will be able to…

- Translate a set of instructions into a form that a computer can carry out.
- Extend or modify a program written by another person to change the programs behavior to meet a slightly different use case.
- Implement various features and strategies in writing a program, including conditional logic; iterative control structures, and variables.

5c. Creating Abstractions

Abstraction is the ability to represent an idea or a process in more general terms by foregrounding the important information while backgrounding unnecessary aspects or details of the problem. Abstraction is an important technique in solving problems that are similar in nature. Creating abstractions is a powerful way to take advantage of computational power to solve STEM problems.

Students will understand that…

- Abstraction is the process of pulling out the important information, while reducing unnecessary details of an idea or a procedure. This allows you to use the abstracted idea in the context it was created as well as new, similar situations.
- In a computer program, there are many different abstracted concepts that you can use, including variables, objects, classes, and functions.
- When using computational abstractions in a computer program, it is often useful to name them in a way that properly represents the properties of that concept.
- The strength of a computational abstraction is its ability to apply to a variety of situations.

Students will be able to…

- Identify a set of useful abstractions that can yield advances in or solutions to a particular problem.
- Explain the relationship between a set of computer-based abstractions and their real world analogs.
- Use a set of provided computer-based abstractions to solve a problem.
- Use abstracted concepts in computer code or a pseudo-code, including variables; objects; classes; and modules.

5d. Applying Conditional Logic

Conditional statements give the user the ability to define under what conditions a certain set of instructions should be executed. Conditional logic is a key strategy for introducing logical breaks into computational environments in STEM fields.

Students will understand that…

- Conditional statements are used to control which instructions are executed in a given run of the program.
- Conditional statements make it possible to not have every step of an algorithm be executed on every run.
- In conditional statements, the "condition" is an expression that evaluates to either

true or false (Boolean expression). The value of the condition determines what set of instructions are executed.

- Along with the value of the condition, the state of the program at the time of its execution determines which instructions to execute (by state we mean the value of the variables defined in the program). If the values of items in the conditional statement are changed later in the program, the conditional statement is not reevaluated.
- Different programs/computer languages represent conditional logic in different forms. Despite differing appearances, the different forms are conceptually equivalent.

Students will be able to…

- Use conditionals in an algorithm, wherever conditional control logic is needed.
- Define and write valid conditional statements to create specified behaviors in a program.
- Trace an algorithm, identify which steps are executed, and predict the output of a computation based on the arrangement of conditional statements.
- Think through "what if" scenarios and test what the output of a system would be if its variables and/or inputs are altered.

5e. Effectively Using Recursion and Iterative Logic

When solving a problem, it is often necessary to repeat a set of steps multiple times to achieve a desired result. This can be done both recursively or through an explicitly defined number of iterations. In STEM fields, iterative logic and recursive procedures are commonly used when working with large data sets or developing computational solutions to problems.

Students will understand that…

- Iterative logic causes steps of an algorithm to be repeated.
- An iterative block repeats either by a fixed number of times, or as long as the value of a conditional test remains true.
- Iterative logic changes the flow of execution in a program. The code inside the iteration block is executed repeatedly based on the logic controlling the block. Upon completion, execution continues with the lines of code following the iterative block.
- There are different types of control structures you can use to introduce iterative logic into a program, such as *while* loops and *for* loops.
- Recursion is similar to iteration, but repeats instructions in a self-similar manner. In other words, in a recursive function, the instructions are executed repeatedly by having the function calling itself.

Students will be able to…

- Identify the presence of iterative logic or recursion in an algorithm and explain the effects on the execution of an algorithm.
- Identify how an iterative approach to a problem could express the problem more concisely; and explain how this approach could be used in an algorithm.
- Implement a proper form of iterative logic when writing a computer program.
- Define recursive loops properly and implement them in a computer program.