Understanding Systems with Computational Models: Creating and Assessing a Computational Model

A system is a group of things which affect each other, such as plants and animals in a food web or parts of a machine. Models and simulations represent relationships and processes of systems with interrelated parts. Models can be computational or non-computational. **Computational models** represent mathematical relationships between parts of a system and are created using a computer. In this activity, you will create a computational model that represents a real world system.

1 Part 1: Identifying a Problem

Computational models can help us to understand real-world phenomena that are difficult to observe because of size, time, and/or visibility.

- Identify a real-world process or problem that a computational model might help you to illustrate.

- How does this phenomenon relate to your life and/or community?

- What do you wonder about this phenomenon?

- How can this model help you to better understand this phenomenon?

Digital Promise
Accelerating Innovation in Education

Sketch and label the phenomenon you will model here:

Digital Promise
Accelerating Innovation in Education

Think about the parts of the phenomenon, the purpose of each part, and if the part will perform an action in your program. Then, consider if the action depends on a variable. If it does, describe how the part, action, and variable are related. Here, you are describing the mathematical relationships you may illustrate in your model.

Parts Objects in the model	Purpose Role of the object in the system	Action Something the object might do	Variable A characteristic of the object that may determine if/how the action occurs	How does the variable determine whether the object will do the action? Describe the relationship or equation.
Ex: Lake	*Ex: storing water*	*Ex: Evaporate*	*Ex: Temperature*	*Ex: If the temperature is greater than 100 degrees C, the lake will evaporate*

A model is a type of <u>abstraction</u>, or reduction of something to a very simple set of characteristics. In the last column of the chart, circle the relationships between parts, actions, and variables that are essential to understanding the purpose of the model you described in Part 1. Explain below:

_____

_____

_____

_____

3 Part 3: Drafting Your Computational Model

Now you will use a computational tool to create your computational model. There are many tools available to create computational models, such as coding platforms (e.g., Scratch, Snap, StarLogo Nova) or computational modeling platforms (e.g., SageModeler, Insight Maker, Loopy). Your teacher will tell you which tool(s) you may use for this assignment.

4 Part 4: Pair Debugging and Assessing Your Computational Model

(Note: Use this step if coding was used to create the model.)

Review the code created for this model. Try an input and observe the results. Does the result match what was supposed to happen?

No →

Talk with the partner who developed this model to share your process and feedback:

• What input did you use?

• How was the intended output different than what actually happened?

Use this feedback to fix the identified issues then go back and start with a new input. If there are no more inputs to test, go to the next questions.

Yes ↓

Think about the efficiency of the code. Do you observe any repeating patterns in the procedure?

No →

Complete using and modifying computational models.

• What changes can you make to this model to make it more similar to the real world or better inform your question?

Use this feedback to improve the model.

Yes ↓

Can this code have the same result with less steps?

No →

• Have a new partner use this flowchart to work through your code.

• Repeat this debugging algorithm until two partners are able to complete your code without errors.

Yes ↓

Talk with the partner who developed this code to share your process and feedback:

• Where do you observe patterns in the code?

• In what ways, could you modify the code to be more efficient?

Use this feedback to fix the identified issues.

• Have a new partner use this flowchart to work through your code.

• Repeat this debugging algorithm until two partners are able to complete your code without any errors.

Digital Promise
Accelerating Innovation in Education