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?

How does a submarine not sink or float (remain neutrally buoyant) in water? What factors determine if a submarine will sink or float?

A model can help me understand how a submarine remains neutrally buoyant by demonstrating the effect of variables on a submarine’s buoyancy.
Sketch and label the phenomenon you will model here:

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.
<table>
<thead>
<tr>
<th>Parts</th>
<th>Objects in the model</th>
<th>Purpose</th>
<th>Role of the object in the system</th>
<th>Action</th>
<th>Something the object might do</th>
<th>Variable</th>
<th>A characteristic of the object that may determine if/how the action occurs</th>
<th>How does the variable determine whether the object will do the action? Describe the relationship or equation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex: Lake</td>
<td>Ex: storing water</td>
<td>Ex: Evaporate</td>
<td>Ex: Temperature</td>
<td>Density of the lake</td>
<td>Ex: If the temperature is greater than 100 degrees C, the lake will evaporate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submarine</td>
<td>Traveling in the water</td>
<td>The submarine is going to sink, float, or neither in the water.</td>
<td>Density of the submarine</td>
<td>The mass of the submarine: • What it’s constructed of • How many people/how much &quot;stuff&quot; it’s carrying</td>
<td>The volume of the submarine</td>
<td>Density = mass/volume. The volume of the submarine remains constant. The submarine can add mass by increasing the amount of water in its ballast tanks. This increases the density of the submarine. If the density is greater than the density of the water, it will sink. The submarine can lose mass by decreasing the amount of water in its ballast tanks, filling with air instead. This decreases the density of the submarine. If the density is less than the density of the water, it will float.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>Medium for submarine to travel</td>
<td>Might move/have a current • Might be more or less dense based on the depth</td>
<td>Density of water</td>
<td>Temperature of the water</td>
<td>Salinity of the water</td>
<td>Motion of the water</td>
<td>Depth of the water</td>
<td>The density of water is 1 g/mL. The density of water changes with temperature and salinity. Ocean currents move objects. Deeper water can be more dense due to increased pressure.</td>
</tr>
<tr>
<td>Sunlight</td>
<td>Affects water currents and temperatures</td>
<td>Might warm shallow water • Might cause a current</td>
<td>Hours of sunlight</td>
<td>Intensity of sunlight</td>
<td>Changes in water temperature cause convection currents.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ocean floor features (subduction zones, rifts)</td>
<td>Affects ocean currents and temperatures</td>
<td>Areas of thinner crust could warm the water.</td>
<td>Depth of crust</td>
<td>Temperature of water near crust</td>
<td>Changes in water temperature cause convection currents.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A model is a type of abstraction, 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:

In order to explore how a submarine sinks or floats in water, it is most important to determine the density of the submarine compared to the density of the water. If the submarine is more dense than water, it will sink. If it is less dense, it will float.

The density of the submarine is determined by mass/volume. There are factors that affect mass and volume that are determined when the submarine is constructed, such as how big it is and what it is constructed of. These will be excluded from our model because they are not easily modified. The mass of the submarine can be modified by adding people to the submarine or adding water to the ballast tank. This will be a variable that can be modified in the model.

There are factors that can modify the density of water (e.g. salinity, temperature, sunlight, ocean floor features), but these variations are slight and therefore we will exclude from the computational model, keeping the density of water at a constant of 1 g/mL.

### 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.

[Link to Scratch model here](#)
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?

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.

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

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 errors.

Can this code have the same result with less steps?

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.
- 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.