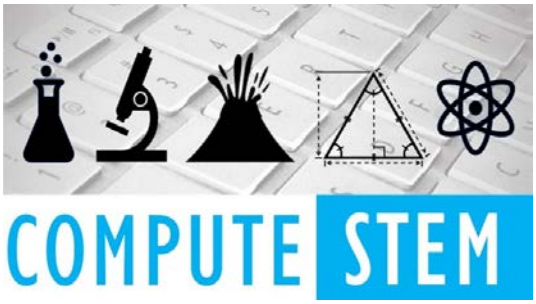


Specific Heat Capacity Coding Lesson by Le and Ermer

Objectives:
SWBAT rearrange the specific heat capacity equation to solve for different variables
SWBAT create a program which will take in variables as inputs by analyzing a similar code
SWBAT use their created program to solve various problems associated with specific heat capacities.

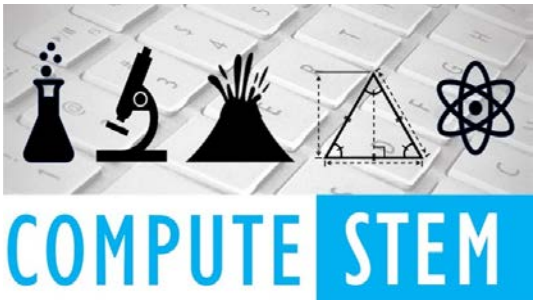
Engage
Opening question - If you could have a robot do anything you wanted, what would it be?
This question is then tied to the lesson, coding the specific heat capacity of an equation. Explain that programming is especially powerful when it comes to math, and it can greatly help students master the concept rather than bumbling through the math.

Explore
Pair students into groups of two with a laptop, and have them open the code. The full code is in this doc, but students will only be able to see two of the five functions.
Ask students to analyze what the code does in their own words. Student responses may include "Calculates a specified variable," "Does the math for me," etc.
Ask students if there is anything missing in the program that could make it better? Can you make those changes?

Explain
Allow students time to complete the other three functions. Go around and monitor student progress. Encourage students to keep trying if their code doesn't work the first time around. Students would be provided with a list of resources that may help them if they run into difficulty.

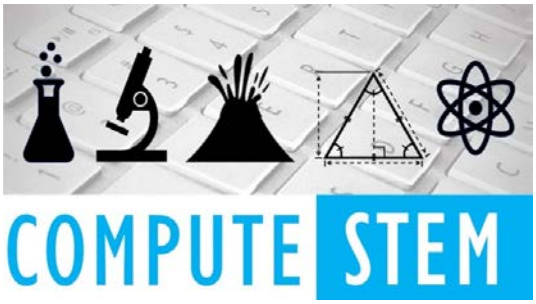
Elaborate
Allow students time to think about any other equations they could code in Python that would be useful to them in Chemistry (E = hv, m1v1 = m2v2, PV = nRT, etc.)
Challenge students to try and code it based on the given code. Students will be given an opportunity to share what the code that they created or attempted.

Evaluate
Exit ticket - various problems related to specific heat capacity that they will attempt without their program and then they can check their work using the program.

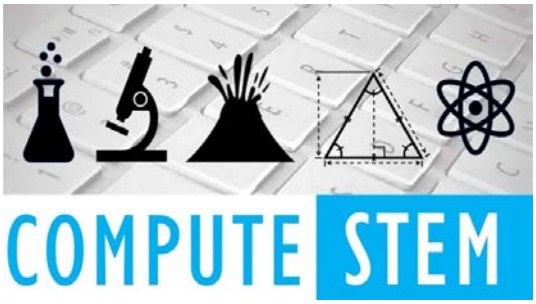


Code

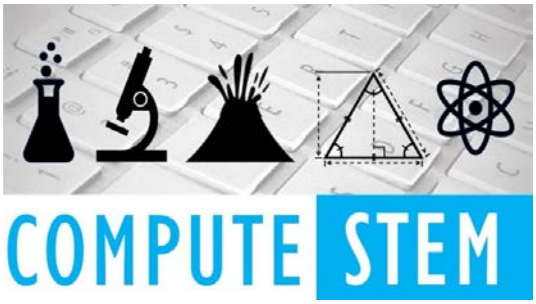
```
specific_heats = {"H2O": 4.184,
                  "NH3": 4.70,
                  "Al": 0.902,
                  "C": 0.720,
                  "Fe": 0.451,
                  "Cu": 0.385,
                  "Au": 0.128,
                  "C2H5OH": 2.46,
                  "C2H6O2": 2.42,
                  "CCl4": 0.861,
                  "CCl2F2": 0.598}

#variables: q, m, cp, ti, tf

questionlist = ["What is the value of q in joules? ",
                "What is the value of m in grams? ",
                "What is the value of cp in J/g K? ",
                "What is the value of ti? ",
                "What is the value of tf? ",
                "What is the chemical formula of the substance?"]


def q():
    m = float(input(questionlist[1]))

    cp = input(questionlist[5])
    if cp in specific_heats:
        cp = float(specific_heats.get(cp,""))
    else:
        print ("This chemical is not in the database.")
        cp = float(input(questionlist[2]))

    ti = float(input(questionlist[3]))
```




```
    tf = float(input(questionlist[4]))
    dt = tf - ti
    q = m * cp * dt
    print (q)

def m():
    q = float(input(questionlist[0]))

    cp = input(questionlist[5])
    if cp in specific_heats:
        cp = float(specific_heats.get(cp,""))
    else:
        print ("This chemical is not in the database.")
        cp = float(input(questionlist[2]))

    ti = float(input(questionlist[3]))
    tf = float(input(questionlist[4]))
    dt = tf - ti
    m = q / (cp * dt)
    print (m)

def cp():
    q = float(input(questionlist[0]))
    m = float(input(questionlist[1]))
    ti = float(input(questionlist[3]))
    tf = float(input(questionlist[4]))
    dt = tf - ti
    cp = q / (m * dt)
    print (cp)

def ti():
    q = float(input(questionlist[0]))
    m = float(input(questionlist[1]))

    cp = input(questionlist[5])
    if cp in specific_heats:
        cp = float(specific_heats.get(cp,""))
    else:
        print ("This chemical is not in the database.")
```




```
        cp = float(input(questionlist[2]))

    tf = float(input(questionlist[4]))
    ti = tf - (q / (m * cp))
    print (ti)

def tf():
    q = float(input(questionlist[0]))
    m = float(input(questionlist[1]))

    cp = input(questionlist[5])
    if cp in specific_heats:
        cp = float(specific_heats.get(cp,""))
    else:
        print ("This chemical is not in the database.")
        cp = float(input(questionlist[2]))

    ti = float(input(questionlist[3]))
    tf = (q / (m * cp)) + ti
    print (tf)

def loop():
    question = input("Which variable are you solving for? Your choices are q, m, cp, ti, or tf. ")
    question = question.lower()
    if question == "q":
        q()
    elif question == "m":
        m()
    elif question == "cp":
        cp()
    elif question == "ti":
        ti()
    elif question == "tf":
        tf()
    else:
        print ("That is not a valid variable. Please enter a valid variable again.")
        loop()
loop()
```


COMPUTE STEM