TEACHER Worksheet: Build-your-own Valence Finder

Subject: Chemistry & Computational Thinking **Grades levels:** 6 - 12

Description: By building a program to determine the valence of ANY element on the first three rows of the Periodic table, students learn the steps to solve the problem while learning how to program logic and think about processing data in sequence. NOTE: The worksheet includes the option of letting students create a bug that they have to fix.

• Uses Python 2.7 programming language interface, which needs to be installed on computers. Python Download and Install Instructions. Python intro.

• Program example file link.

 This icon is used for teacher suggestions throughout the lesson. Answers to the questions will be inline in **red**.

 Prior to conducting exercise with a class:

1. The (free) python programming interface needs to be installed on all the computers. It's available at http://python.org/download/
2. The instructor should work through the worksheet to become familiar with the IDLE shell window and the program writing window.

What is valence? – some atoms are "happy" alone and don't care to play or bond with others. Some atoms are "unhappy" alone and really want to play (bond) with others. What determines whether an atom is happy or unhappy and if it wants to form bonds?..... The number of electrons it has in its outer ring/shell in relationship to number possible in that ring/shell…….valence!

For the purposes of determining valence, it is useful to think of the electrons of a particular element as being arranged in layer or shells or rings. The first shell, located closest to the nucleus, can hold 2 electrons. The next two shells can hold 8 each. The electrons fill in the closest shells first and any left over fill the next shell out.

Whenever you see a ?, partners should write out answers **together**. If you don't know the answer, ask another group for help before moving on.

Every atom of an element *usually* has the same number of electrons as any other atom of that element. And most elements usually have the same number of electrons as protons. The numbers of protons (and electrons) in an element is known as its atomic number, and this number is different for

each element. The atomic number for fluorine (F on the periodic table) is 7, which means it usually has 7 protons and 7 electrons.

Now let's figure out it's valence.

1. How many electrons are on its first shell? 2
2. How many electrons are on its second shell? 5
3. Are there any electrons on its third shell? None
4. Which shell is the "outer shell"? The second
5. How many electrons can the outer shell hold? 8
6. If having a full outer shell is what will make an element "happy" what does fluorine want to do to become happy? (hint: it can do two things, but one is more likely)

It wants to gain 3 electrons

The number of electrons an atom/element wants to give or get to become happy is its *valence number*.

7. What is the valence number for fluorine? 3

Valence also has a sign (for charge) which matches what charge the atom will be if it loses or gains electrons.

8. If an atom gains an electron or electrons, will it be negatively charged or positively charged – will it have more electrons than protons or less electrons than protons? Negatively charged
9. What is the valence charge for fluorine? - 3

The combination of an element's valence number and valence charge IS its *valence*!

For this exercise we will be creating a computer program to determine valence, for any element in the first three rows (periods) of the periodic table. As we work through a valence problem together, we will also write code in the program to do the calculations for us. After we finish writing all the code and save our program, all we need do is run it and enter the value for an element's atomic number and the program will do the rest!

The programming language we will be using is Python. Don't be scared, it won't bite!

What is Python?

Python is a programming language. It is used to make programs. The programs make the computer do stuff. Computers are stupid but follow directions very well … and fast! Everything that one does with/on a computer or the internet happens within a program that someone created.

Python commands to create an interactive program:

=	sets or resets the value for a variable. For example, "x = 5" sets a value for the variable x as 5.
==	tests to see if true
print	this is needed in **a programming window** so that the the computer will display whatever comes after. For example, "print eval('35+24')" tells Python to display the answer to 35 + 24.
input	allows the program to ask a question of the user and record the answer. For example, "c=input('How many pets do you have?')" will ask the user for his or her number of pets and store the number entered as the variable c.
if	conditional statement to test that a condition has been met
elif	next in a series of alternative conditional tests. Used after "if".
else	last in a series of alternative conditional tests. Used after "elif" or "if".

Procedure:

The first thing we need to do is open up the shell window. Your teacher will tell you where to find the icon for the application called IDLE. Open IDLE to get to the shell window. The shell is where we we eventually see our program run.

Whenever you see a, , one partner should be working on the computer.

The next thing we want to do is open a new programming window. Go to the File menu and drag down to New Window. The programming window is where we will be writing our code. NOTE: the way to easily tell the difference between the shell and prograqm window is that the shell will have the symbol >>> that precedes the blinking cursor and the programming window will be completely blank.

Since we want our program to be a calculator that can use ANY atomic number let's make it ask for one by typing the following in the new program window:

n = input("What is the element's atomic number?")

What this does:
asks the user for input, which it will store in a variable named *n*.

The value will be stored in the variable and could be used by other equations or it can be displayed by…this code (type it on the next line):

print n

Now let's save our program as *ValenceFinder.py* and the go to the *Run* menu and choose *run module*. NOTE: The program cannot be run until it is saved first. When it asks you "What is the element's atomic number?", type it in and push return.

 Make sure they put .py at end.

Now let's make the program find the valence for elements on the **first** row of the periodic table.

10. What is the atomic number for Hydrogen? 1
11. How many electrons does Hydrogen have? 1
12. What is the atomic number for Helium? 2
13. How many electrons does Helium have? 2
14. Why do you think these are the only two elements on the first row? (hint: think about shells) They only have electrons on the first shell (which can hold only 2 electrons)

For this section of our program we want to deal with only elements with these two atomic numbers. How can we specify only these two possible numbers: 1 or 2? If n<3

Let's code it! In the programming window, erase *print n* and type: *if n<3:*

To have the program deal with hydrogen, whose atomic number is 1… on the next line, indent with one push of the tab key and type *If n==1:* and push return to get to next line (note: putting a colon after the if statement will help it automatically indent). On this line code*: print "I'm not happy. I have one electron in a two-possible shell. My valence is positive or negative 1 electron. I'd be happy to give or get one"*

To have the program deal with helium, whose atomic number is 2… on the next line, indent with one press of the tab key and type *else:* and on the next line: *print "I am very happy. I have two electrons in a two possible shell. I don't want to give or get any electrons."*

Now lets make the program find the valence for the elements on the **second** row of the periodic table. For this section of our program we want to deal with only elements with two atomic numbers between 3 and 10 (inclusive).

15. Why do we only go up to 10 in this row? They only have electrons on second shell (plus 2 on the first)
16. How can we specify ONLY numbers in this range? elif 2<n<11

What this does:
This specifies a range of atomic numbers for the second row which will be dealt with by code written next.

Let's code it! In the programming window, on the next line, backspace (twice) to back up to line up with the first if statement, so that we can specify an alternative to *if n<3* and type: *elif 2<n<11:*


Make sure they backspace to outdent.

17. If we know how many total electrons our elements has, how can we get rid of the electrons on the first shell in order to determine how many are left over for the second shell? v=n-2

Let's code it! On the next line type: *v=n-2*

There are eight possible electrons on this shell. Only one is "happy", but there are three possible types of "unhappy".

18. Can you think of what the three possible types of unhappy elements there could be? Ones that want to gain electrons, lose electrons and one that would be happy to gain or lose.

Now, let's think about the one "happy" element…….

19. If we want the program to deal with the one “happy” valence element, we need help the program to recognize it. How many electrons would a “happy” element in this row have in its outer shell? 8
20. How can we write a conditional if statement to verify that? v==8

NOTE: We CANNOT use *if v=8* to test what is currently held in v because *v=8* would put 8 in v no matter what had currently been there. We CAN use *v==8* which test to see if it is TRUE. We also need to put a colon after the test to signal an action is going to be performed – in this case

Code it on the next line.

And put this on the next line: *print "I am very happy. I have 8 electrons in an eight-possible outer shell."*

Let's think about one of the three "unhappy" types elements that will <u>want to give away</u> electrons (which will give them a positive valence)…….

21. If we want the program to recognize the “unhappy” **positive** valence elements, we need help the program recognize them. How many electrons would a positive valence “unhappy” element in this row have in its outer shell? There are several possiblities…v=1, 2 or 3
22. How can we write a conditional if statement to verify <u>any</u> of those possibilities? v<4

Code it on the next line. Use *elif* and a : as this is the second conditional test in a series. AND backspace once to outdent to line up with *if v==8:*


It is very importantsto get the quotations and commas correct in the print code.

Put this on the next line: *print "I'm not happy. I have", v, " electrons in a eight-possible outer shell. My valence is plus", v, "electrons. I want to give", v," electrons away and when I do I will be more positively charged."*

Let's think about another of the three "unhappy" types of elements that will want to gain

> NOTE: The , *v,* inserted a various point in between quoted text will tell the program to print the value held in the v variable, at those points in the printed text. Make sure the commas and quotation marks are exact.

electrons (which will give them a negative valence)…….

23. If we want the program to recognize the "unhappy" **negative** valence elements, we need help the program recognize them, too. How many electrons would a negative valence "unhappy" element in this row have in its outer shell? There are several possiblities…v=5, 6 or 7
24. How can we write a conditional if statement to verify any of those possibilities? v>4

Code it on the next line. Use *elif* and a : as this is another conditional test in a series. Don't forget to backspace once to outdent to line up with *if v>4:*

And put this on the next line: *print "I'm not happy. I have", v," electrons in a eight-possible outer shell. My valence is negative ", 8-v,". I want to get ", 8-v," electrons and when I do I will be more negatively charged"*

Let's think about the last of the three "unhappy" types of elements. It would be happy to lose or gain electrons (which would give it positive OR negative valence)…

25. If we want the program to recognize with the "unhappy" valence elements, we need help the program to recognize it. How many electrons would a positive OR negative valence "unhappy" element in this row have in its outer shell? 4
26. How can we write a conditional if statement to verify that? v==4

Code it on the next line. Use *elif* and a : as this is another conditional test in a series. AND backspace once to outdent to line up with *if v==8:*

And put this on the next line: *print "I'm not happy. I have 4 electons in a eight-possible outer shell. My valence is positive AND negative 4. I want to get OR give 4 electrons.."*

We're done for the second row possibilities. You can save and run it. Your program should look like this

```
n = input("What is the element's atomic number?")

if n<3:
    if n==1:
        print "I'm not happy. I have one electron in a two-possible shell.  My valence is plus or minus 1 electron. I'd
    else:
        print "I am very happy.  I have two electons in a two possible shell.  I don't want to give or get any electrons
elif 2<n<10:
    v=n-2
    if v==8:
        print "I am very happy. I have 8 electrons in an eight-possible shell."
    elif v<4:
        print "I'm not happy. I have", v, " electrons in a eight-possible outer shell.  My valence is plus", v, "electro
    elif v>4:
        print "I'm not happy.  I have", v," electons in a eight-possible outer shell.  My valence is negative ", 8-v,".
    elif v==4:
        print "I'm not happy.  I have 4 electons in a eight-possible outer shell.  My valence is positive AND negative 4
'
```

27. What do you think would be a quick way of creating code for the third row? Copy and paste

Outdent all the way, before you paste.

Recycling code by copying and pasting is done often. It saves lots of typing time!

28. What do we need to change here *elif 2<n<11*: ? Change to *elif 10<n<19*:

Code it. Save and run it, using a (atomic) number for any element in the third row.

29. What's wrong? Need to change *v=n-2* to this *v=n-10*

 This copying/coding error was included to illustrate what can happen if a line of code is not correct and the need to "debug" it. It can be included or changed (by inserting the correct instruction in the worksheet) to eliminate it.

We need to fix something in the programs code. This is called de-bugging.

30. What do we need to fix? Need to change *v=n-2* to this *v=n-10*

Code it. Save and run it, using a (atomic) number for any element in the third row?

Now the last thing we need to do is write a line that will deal with numbers beyond the third row. We don't really need to test or set up an *if* condition. All we need to do to catch all the atomic numbers we have not already recognized is outdent all the way and put *else:*

And put this on the next line *print "I'm not on the first three rows so you don't need to know now."*

Code it. Save and run it!

In our program we had to tell the computer how to recognize and indentify valence for any of the elements in first three periods of the table.

31. How did we initially set up the section that dealt with ONLY items on the first row (the code)? if n<3:

32. The second? elif 2<n<11:

33. The third? elif 10<n<19:
34. How did we enable the program to identify any of several elements in either the second or third row, which would have a negative valence? v>4
35. A positive valence? v<4
36. A positive or negative valence? v==4
37. Why did we not have to go through such elaborate programming for the first row? We had just two possibilities

If you want you can email the file to yourself, so you can use it to check any valence problems you practice at home. To use your program, you'll need the Python shell (IDLE comes in the bundle), which you can download for free at http://python.org/download/

CA 8th Grade Science Standards
Structure of Matter
3. Each of the more than 100 elements of matter has distinct properties and a distinct atomic structure. All forms of matter are composed of one or more of the elements. As a basis for understanding this concept:
a. Students know the structure of the atom and know it is composed of protons, neutrons, and electrons.

The ISTE - National Educational Technology Standards (NETS•S)
1. Creativity and Innovation
Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology. Students:
a. apply existing knowledge to generate new ideas, products, or processes.
3. Research and Information Fluency
Students apply digital tools to gather, evaluate, and use information. Students:
d. process data and report results.
6. Technology Operations and Concepts
Students demonstrate a sound understanding of technology concepts, systems, and operations. Students:
a. understand and use technology systems.
b. select and use applications effectively and productively.
c. troubleshoot systems and applications.