

## Student Worksheet: Build-your-own Hardy Weinberg Calculator (Extensions)

This worksheet contains 2 extensions:

**"Pre" extension** - The first extension creates programming to ask the user for values that are needed to figure out the frequency of individuals showing the recessive trait. This is THE one number needed to calculate all the other frequencies. The frequency of individuals showing the recessive trait is the only one that can be observed in a population BUT it is not always given in the word problem.

**"Post" extension** - The second will take the frequencies and apply them to an actual population number to generate actual numbers of individuals of the 3 genotypes and numbers of each allele ... in that population.

### 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".

### "Pre" extension procedure:

 Let's open the program we saved called *HardyWeinbergCalculator.py*.

 The next thing we want to do is to Save as to another name HWpre.py, just in case we make some mistake, so we can always go back to the original version.

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

The code will be creating today is designed to calculate the frequency of recessives and put that value into the named variable *freqhh*. Some times word problems don't give you this number, but you might be able to find it using other numbers the problem does give. The first thing we want to have our program do is ask for any of those other numbers, and to store them in a variable we name. Then, depending on what numbers the program is given, it will perform a different calculation to find the key value - the frequency of recessives - and put that value into the named variable *freqhh*.

 The program already asks for the frequency of recessives with the code: *freqhh = input("What is the frequency or percent of individuals displaying the recessive trait which, BTW, must be homozygous recessive?")* but we can't put anything in the variable *freqhh* because the problem might not give that value and we might have to determine what to put in another way.... So delete that line and put: *rec=input("What is the frequency showing the recessive trait? Put zero if you don't know.")* The variable called *rec* will be a temporary place to put a value if the problem gives it. We have to also ask for zero if the problem doesn't give it, because we may need to evaluate a numeric value for this variable for later choices.

📖 On the next line, to have the program ask for the frequency of individuals displaying the dominant trait type: `freqdom=input("What is the frequency showing the dominant trait? Put zero if you don't know.")`

📖 On the next line, to have the program ask for the number of individuals displaying the dominant trait type: `numdom=input("What is number of individuals showing dominant trait? Put zero if you don't know.")`

📖 On the next line, to have the program ask for the number of individuals displaying the recessive trait type: `numrec=input("What is number of individuals showing recessive trait? Put zero if you don't know.")`

📖 On the next line, to have the program ask for the total population size type: `tot=input("What is total population? Put zero if you don't know.")`

Your code should look like this:

```
from math import *
```

---

```
rec=input("What is the frequency showing the recessive trait? Put zero if you don't know.")
freqdom=input("What is the frequency showing the dominant trait? Put zero if you don't know.")
numdom=input("What is number of individuals showing dominant trait? Put zero if you don't know.")
numrec=input("What is number of individuals showing recessive trait? Put zero if you don't know.")
tot=input("What is total population? Put zero if you don't know.")
```

📖 We are now going to have the program test to see whether the value for the frequency showing the dominant was given by testing for something other than zero, using the *if* conditional statement by typing on the next line: `If freqdom>0:`



If we (and the program) know this number, how do we use it to calculate *freqhh*? \_\_\_\_\_

📖 Code it on the next line.

📖 On the next line we will write a conditional if statement to see if the number of individuals showing the recessive trait was given. Code it.

NOTE: The conditional test statement *elif* should be used if there will be a series of conditional tests (there will be)

📖 On the next line we will use *numrec* to calculate *freqhh* (we will assume that if this value was given in the word problem, the total population number was given as well). Code it.

📖 On the next line we will write a conditional if statement to see if the number showing the dominant trait was given. Code it.

📖 On the next line we will use *numdom* to calculate *freqhh* (we will assume that if this value was given in the word problem, the total population number was given as well). Code it.

📖 The last condition will only need to use *else*: Code it on the next line.

📖 And then, just in case the frequency of recessives was given, on the next line put `freqhh=rec`.

We're done! Our original code (following) can now work with *freqhh* no matter how it has been derived.

---

---

**"Post" extension** - The second extension will take the frequencies that the calculator calculates and apply them to an actual population number in order to generate actual numbers of individuals of the 3 genotypes and numbers of each allele ... in a specific population.

In order to apply the frequencies or percentages that our Hardy Weinberg Calculator will generate to an actual population we need to ask the user for the actual population. Note: If you did the "Pre" extension, you have already done this. Otherwise.....

📖 After the line `freqhh = input("What is the frequency or percent of individuals displaying the recessive trait which, BTW, must be homozygous recessive?")`, push return to generate a new blank line and type:  
`tot=input("What is total population,? Put zero if you don't know.")`

Most computer programs process information in sequence, therefore if we want to have use the values the program it will calculate for **p, q, 2pq, p<sup>2</sup> and q<sup>2</sup>**, for new calculations of actual population numbers, we have to put that new code AFTER the initial frequency calculations.

📖 Since most of our code to generate actual population numbers (and the print text that results) will look similar to the code used to generate frequencies, you can copy that block and paste it underneath itself.

📖 The first thing to do with the pasted block of code is erase `q = sqrt(freqhh)`. In its place we will do is test to see if a value for the total population for a population was given using the code: `if tot > 0:`

🔍 If we were given an actual population number (now in `tot`) AND the calculator we used the other day has the frequency of the recessive allele (now in `q`) how do we apply that frequency/percentage to the actual population number? \_\_\_\_\_

Hint: remember each individual in the population has TWO alleles!!!!

📖 When we want to print and execute a simple equation we can do it on one line. Instead of `print q, "is the frequency of the recessive allele"` change it to: `print q*tot*2, "is the number of the recessive allele"`

📖 Delete the next line `p=1-q` and all the other lines in this pasted code block which just contain equations. All our equations will be included and executed on the print lines. Like the one we just modified.

📖 On the next print line instead of `print p, "is the frequency of the dominant allele"` change it to: `print p*tot*2, "is the number of the dominant allele"`

📖 On the next print line instead of: `print freqHH, "is the frequency of the homozygous dominant individuals"` change it to: `print freqHH*tot, "is the number of homozygous dominant individuals"`

📖 On the next print line instead of `print freqHh, "is the frequency of the heterozygous dominant individuals"` change it to: `print freqHh*tot, "is the number of heterozygous dominant individuals"`

📖 On the next print line code: `print freqhh*tot, "is the number of homozygous recessive individuals"`

📖 Then, on the next line code: `else:`

📖 Push return and make sure it is indented (it should have if you put the colon after else) : `print "Done"`

Your code should look like this:

```
tot = input("if you want to have actual genotype and allele numbers for an act  
if tot>0:  
    print p*tot*2, "is the number of the dominant allele"  
    print q*tot*2, "is the number of the recessive allele"  
    print freqHH*tot, "is the number of homozygous dominant individuals"  
    print freqHh*tot, "is the number of heterozygous dominant individuals"  
    print freqhh*tot, "is the number of homozygous recessive individuals"  
else:  
    print "Done"
```

---

NOTE: The code text of the first line was cut off to make the image fit this Word document. It should read:  
*tot = input("if you want to have actual genotype and allele numbers for an actual population enter it here, if not enter zero.")*

Now run the program and try it out several times to make sure it can handle any variable or variables you give it, which a word problem might have.



Build-your-own Hardy Weinberg Calculator extension lesson by Mark Wenning is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/).