# CSCI 150: Exam 1–Take Home

Due at the beginning of class
Wednesday, February 15, 2023

Please turn in a single .ipynb Kaggle notebook, uploaded to our class TEAMS page. Each function should be in its own cell. Please call your file **Exam1LastName**. This makes grading much easier on me, rather than having 30 files all called "CS Exam Thing."

The *only* resources you may use are the following:

- any code you have created for class, including homework and labs
- any notes you have taken during class
- code and any other material your instructors have posted to either the lecture or lab Teams pages
- any information directly linked from the class homepage, `https://hendrix-cs.github.io/csci150/`
- anything in the official Python documentation, `https://docs.python.org/3/`

You may not talk to a classmate, friend (real-life or Facebook), Siri, or anyone other than me about this exam until you turn it in, nor search the Internet or library or any reference other than those listed above for assistance. You may not even mention anything about how long it took you to complete the exam, that you found problem #2 particularly difficult (or easy), or in fact talk at all about the exam or Computer Science with anyone other than your instructor from Monday, 8:10am - Wednesday 8:10am. Anyone who *gives* answers is equally in violation of the Academic Integrity Policy as one who *receives* them. All suspected violations will be reported to that committee.

1. You might remember from geometry that a triangle with side lengths $a$, $b$, and $c$ is a right triangle when $a^2+b^2=c^2$ (i.e. the Pythagorean Theorem). Write a function `right` which takes in three parameters, `a`, `b`, and `c`, each of which is a float, and returns `True` exactly when the three numbers are the sides of a right triangle (with $c$ being the hypotenuse).

   For example:

   `right(3.0, 4.0, 5.0)` will return `True`

   `right(5.0, 6.0, 7.0)` will return `False`

   `right(5.0, 12.0, 13.0)` will return `True`

   `right(5.0, 4.0, 3.0)` will return `False`

   `right(1.5, 2.0, 2.5)` will return `True`

2. Write a function `smallest` which will take in three numerical values `x`, `y`, and `z` – all of which are `int` – and returns the smallest value (as an `int`).

   For example:

   `smallest(5, 7, 10)` should return `5`

   `smallest(5, 11, 3)` should return `3`

   `smallest(-3, -5, -7)` should return `-7`

   `smallest(5, 5, 2)` should return `2`

   `smallest(5, 5, 8)` should return `5`

   `smallest(5, 5, 5)` should return `5`

3. Write a function `prod_or_sum` which takes in three parameters: two integers, `a` and `b`, and a boolean `prod`. When `prod` is `True`, the function should return the product of *a* and *b* - that is, it should multiply them together and return that answer. Otherwise, it should return the sum – that is, what you get when you add them.

   For example:

   `prod_or_sum(3, 7, True)` should return `21`

   `prod_or_sum(3, 7, False)` should return `10`

   `prod_or_sum(5, -1, True)` should return `-5`

   `prod_or_sum(5, -1, False)` should return `4`

   `prod_or_sum(-4, 4, True)` should return `-16`

   `prod_or_sum(-4, 4, False)` should return `0`

4. Evelyn is writing a function to classify animals that she may encounter in the wild, to decide whether she should be their friend or run away. In particular, the function should decide whether an animal is friendly or not based on its size and color. `size` is an integer parameter indicating the animal's size, and `color` is a lower-case string parameter indicating the animal's color.

   Small animals (those with a size smaller than 10) are friendly. Big animals (with a size greater than 24) are never friendly. Medium animals (with a size between 10 and 24, inclusive) are friendly only if they are yellow or blue. Finally, however, red animals are never friendly, no matter their size. (Note that animals can have other colors besides red, blue, and yellow.)

   Write a function `is_friendly` which takes two parameters, `size`, an integer, and `color`, a string, and returns `True` when the animal is friendly and `False` otherwise.

   For example:

   `is_friendly(17, ’blue’)` will return `True`

   `is_friendly(16, ’red’)` will return `False`

   `is_friendly(20, ’green’)` will return `False`

   `is_friendly(40, ’yellow’)` will return `False`

   `is_friendly(7, ’purple’)` will return `True`