

## **SeriesComputer**

Write a class called "SeriesComputer" with:

- A constructor that accepts a function argument "f" that accepts a positive integer argument "k" and returns a float
- A method "compute" that accepts a positive integer argument "n" and returns the sum for  $k=1..n$  of  $f(k)$
- If `compute()` is called multiple times on the same SeriesComputer object, it should reuse results from previous calls to `compute()` to speed up the computation
- `compute()` should raise/throw an exception if an invalid argument is entered

Next, write a class called "NegativeSeriesComputer" that inherits from SeriesComputer but instead computes the sum for k from -1 to -n of  $f(k)$ .

## **Pie**

Write a class called "Pie" (the baked food) with:

- A constructor that accepts positive float arguments "radius", "height", and "percentage" where radius and height are both in meters and percentage specifies the percentage of the full circular pie that is remaining (initially 100 for a newly baked pie)
- A method "slice" that accepts a positive float "angle" which cuts away a circular sector piece of pie with the specified central angle "angle" (in degrees), returns a new Pie object representing the slice, and modifies the "percentage" variable of the original Pie object to take away the slice from the original pie.
- Both the constructor and the `slice()` method should raise/throw an exception on any invalid input arguments (including if you try to "slice" a bigger angle of pie than is left)

Next, write a class called "SugarPie" that inherits from Pie and has

- An additional constructor argument "sugar\_per\_cubic\_meter" specifying the average quantity of sugar (in grams) per cubic meter of the pie
- An additional method "sugar" that returns the total amount of sugar in the SugarPie object (assuming the pie is in the shape of a perfect disk with the radius and height given in the Pie object)