How to measure the codes using Big O

5 Rules

No	Description	Complexity
Rule 1	Any assignment statements and if statements that are executed once regardless of the size of the problem	O(1)
Rule 2	A simple “for” loop from 0 to n (with no internal loops)	O(n)
Rule 3	A nested loop of the same type takes quadratic time complexity	O(n^2)
Rule 4	A loop, in which the controlling parameter is divided by two at each step	O(log n)
Rule 5	When dealing with multiple statements, just add them up	

You should be warned that some declarations may include initializations and some of these may be complex enough to factor into the efficiency of an algorithm.

BikeRental Class

This class is used to represent Bike Rental system

Attributes:

- **stock**
  (int) The number of bikes that are available currently in the system
  ex. 100

Methods:

- **display_stock()**
  (int) Displays the bikes currently available for rent in the system
  ex. 100

- **rent_bike_on_hourly_basis()**
  To rent a bike on hourly basis to a customer
- **rent_bike_on_daily_basis()**
  To rent a bike on daily basis to a customer
- **rent_bike_on_weekly_basis()**
  To rent a bike on weekly basis to a customer
- **return_bike()**
  Accept a rented bike from a customer, increase number of available bikes and return a bill

Customer Class

This class is used to represent Customer objects

Attributes:

- **bikes**