* Visibility
  + Purpose: To Find the Acquisition of and Loss of a Signal and pair them together.
  + Inputs:
    - Station Instance
    - Azimuth
    - Elevation
    - Minimum Level of Power Required
  + Contains: code which
    - pairs the AOS and LOS found in the pointing function
    - Creates a unique pair listing for AOS and Satellite number since Satellite pairing code does not create unique pairings for AOS and LOS pairing
      * In simpler terms the AOS and LOS pairings found above would have multiple LOS times for a singular AOS
  + Utilizes:
    - Pointing Function
  + Outputs:
    - AOS and LOS List
      * This list contains
        + Satellite number (Satnum), AOS Time, LOS Time, Minimum Power Level Requirement ( in dBm)
  + Notes:
    - Although Visibility Function is listed as a Level 0 Highest level function here, it is more accurately described as a 0.5 level function since it utilizes components of other Level 0 functions but its output is required for other Level 0
    - This part of the code is difficult to understand but essential the LOS and AOS data gathered from the Pointing() function has intrinsic time dependence, so we can create an accurate list by first looping through each list (AOS and LOS) and creating a pairing every time we see the same Satellite number index and when the Loss of Signal is greater than the Acquisition of Signal. Through this logic we will have multiple LOS times for a singular AOS time which is not wanted. So we then create a unique list which is dependence on a string joining the Satellite number and AOS time.

Level 1- Medium Level Functions/Classes

These Functions largely involve calculations being done to inputted values to output to the higher levels.

* *Station*
  + Purpose: create a Station instance from the inputted string.
  + Location:
    - Inside **User\_Input\_parser\_Call**
  + Inputs: Station Location Str
  + Contains:
    - File parser
    - Assignments
  + Outputs: Station Instance with its characteristics
* **SatListPropagate**
  + Purpose: Propagates a List of Satellites using the TLE file
  + Location:
    - Inside **User\_Input\_paser\_Call**
  + Inputs: TLE file
  + Contains:
    - Creates a Listing of Satellites with their properties defined by class Satellite
    - *Satellite*
  + Outputs:
    - SatList (list)
      * Contains all Satellites Data
* *tracking* 
  + Purpose: Reads data in the Tracking Data file and assigns it to a variable called Tracking
  + Location:
    - **User\_Input\_parser\_Call**
  + Inputs:
    - Tracking file location (string)
  + Contains:
    - Parsing and Assignment code
  + Outputs:
    - Tracking instance
      * Contains all properties read in from file
* *Linkinput*
  + Purpose: creates an object with attributes read in from the Link Input file
  + Location:
    - **User\_Input\_parser\_Call**
  + Inputs:
    - LinkInput string
  + Contains:
    - Readlines and assignment code pieces
  + Outputs:
    - Link Data (object)
      * Contains attributes of read in data
* **Mean\_anomaly\_motion**
  + Purpose: calculates Mean Anomaly Motion and Mean Anomaly at time t, where t is epoch seconds from Satellite’s reference Epoch
  + Location:
    - **Sat\_pos\_velCall**
      * Within Time-Satellite Loop
  + Inputs:
    - Time\_dt
      * Datetime object of the current iterated time
    - Satellite Reference Epoch
    - Satellite’s Mean Anomaly at reference epoch
    - Satellite mean motion at reference epoch
    - Satellite “ndot”- derivative of mean motion
    - Satellite “n2dot”- 2nd derivative of mean motion
  + Contains:
    - Refepoch\_to\_dt
    - Code which calculates