Assumptions:

1. I have assumed that the events would flow in through a json file
2. Individual fields in records are validated already
3. Site visit, order and Image records for a customer can only flow in if there is already a Customer record for the customer.
4. Duplicate records are not allowed

Procedure:

1. Events flowing in from the json file are read, validated, and written to a temporary file. In a real-world scenario, it is much easier to have a staging relational or NoSQL databases. This staging database typically sits between the data sources and the data target, which is the data warehouse.
2. My solution, for the purpose of this exercise, is implemented using flat files and without a staging database. File I/O operations are cumbersome, complex, and slow and this process is not generally used in traditional ETL.
   1. As each event is processed, a STATS record for the customer, which contains metrics such as average visits per week, total revenue, revenue/visit, lifetime value is written. At any given point of time, the stats record will have the most up to date metrics on the customer. This would be a simple and efficient solution given a staging database in the ETL process. Aggregation and updation of stats are of relatively low time and space complexity when performed on a database as opposed to on files.
   2. To get the
3. Ingesting records happens in O(n) time, where n is the number of events ingested. This time may become O(n2) if validations of the ingested events are to be performed because the input file scan has to be performed for each ingested event.
4. TopXSimpleLTVCustomers takes O(nlogn) where n is the total number of records in the input file.

Design:

1. ShutterflyDataChallenge.py: Takes input parameters, validates input, parses json file to dictionary and passes it to Ingest method
2. Ingest.py: Takes events records one by one and passes them to the necessary constructor to create a record in the file. For each SiteVisit and Order record the stats are updated. The method topXSimpleLTVCustomers returns the customerid’s and ltv of the top X customers with highest ltv using the stats record .
3. customer.py: initializes customer record and writes it to the input file.
4. order.py: initializes order record and writes it to the input file.
5. siteVisit.py: initializes siteVisit record and writes it to the input file.
6. Image.py: initializes siteVisit record and writes it to the input file.
7. Stats.py: initializes stats record with fields total visits, average revenue per visit, average visits per week, customer ltv, and writes it to the input file.

Running the code:

* 1. Cd into the folder and enter the following command:

python ShutterflyDataChallenge.py ‘TopXCustomers’ ‘inputfilename’

TopXCustomer should be an integer great than zero and inputfilename should be a string. The input file should be present in the same folder as the