

# Account Score Initiative

## PROBLEM DESCRIPTION

Current account sorting is currently based on last name or account number. Upper management has decided that such sorting is likely not optimal. Management wants to score accounts and allow the the call center to move through accounts top to bottom based on a descending score.

## ASSIGNMENT DETAILS

See attached diagram for data flow details. The database administrator will generate a file every morning at 5 a.m. The programmer must parse the file, score the accounts and place the updated file in the target directory for the account dialer to pick up. The account dialer expects a specific CSV format for proper processing. The technical details are below.

## TECHNICAL DETAILS

- Dialer required format
  - last name, first name, cell, home, account#, score
- Score algorithm
  - $(\text{Pos Contact} * 50) - (\text{Neg Contact} * 100) + (\text{contact} * 10) + (\text{previous payment count} * 100)$   
+ if promise to pay > 3 and no payments  $-(\text{pos contact} * 100)$
- The file for parsing will be named loadYYYYMMDD and the dialer file should be named dialYYYYMMDD
- The automation server will have the locations mapped as drive letters.
- The file class should contain a 'destination' variable that represents where the file should be placed for loading and a 'source' variable that represents the initial file for parsing. Later on, these variables can be changed to production values.

## WORK ASSIGNMENT

### EXPECTED CLASSES

#### File Class:

Reads file data as a dictionary

Writes data to updated file

Use csv module, datetime module

Use the dictreader and dictwriter methods

#### Person Class:

first name & last name

#### Contact Class:

Address & phone #'s

#### Stat Class:

All contact types & other stats  
method(s) that calculate the score

#### Account Class (Composition):

Account Number, Person, Contact, & Stats

### NOTES:

The account class contains many of the previous classes except the file class. Read the file, create a list of the Account class objects that the file class then takes and places in a CSV file at the indicated position.