Human Capital Management Coding Challenge – Bhimesh S. Chauhan
Explanation, Assumptions and Execution
Time Taken: 4 hrs.

Execution:

1. Navigate to codebase folder in bash.
2. Run command in bash: $ pip install requirements.txt
3. Open read_dtf.py file in any ide (preferably pycharm)
4. Run the Code and wait until "Process finished with exit code 0" appears.
5. Navigate to out folder and open Trade_Reconciliation.xlsx file.

There are two sheets in Trade_Reconciliation file. Reconciliation sheet has three columns for Completed Order, Partial Order and Irregularities. The second sheet is called analytics will show detailed statements of each of the six categories where the order was miscalculated and had irregularities. If a column is empty, then there are no such orders in the data provided.

Assumptions:

1. The header in source files stay consistent.
2. Data provided have equal column lengths, meaning that data is provided for all orders.
3. Assuming that there are no repeats in the Trade_Orders file.
4. Assuming that City Column in Trade_Confirms is not important for business analytics.
5. Numbers given are always integer or version of one and not a string.

Explanation:

The following Object-Oriented Approach was taken to complete the task. There are three methods in the code solution with the mentioned cases addressed and functionality.

1. read_data:
   Functionality: To read the data from the file and convert it to dictionary.

2. multiple_execution
   Functionality: To check if there has been two or more than two executions in the Trade_Confirms file and return the list.
   Cases:
   i. Return data with all the duplicates
   ii. Return data omitting the duplicates

3. excessive_execution
   Functionality: To check if more orders have been executed compared to the orders that were made.
   Cases:

i. If Trade_Confirms execution orders are more than Trade_Orders orders.
ii. If Trade_Confirms execution orders are less than Trade_Orders orders.
iii. If Trade_Confirms execution orders are equal to Trade_Orders orders.
iv. If orders are not executed in Trade_Confirms.

4. inconsistent_cost

Functionality: To check if there has been inconsistent cost has been added due to calculation error in tax or commission etc..

Cases:

Method keeps account for current exchange rates.

i. If Currency are different in Trade_Confirms file.

1. If the Total Amount charged does not matches Total Charge accounting for orders on Trade_Orders Page.
2. If Total Amount charged does not matches Total Charge for orders executed Trade_Confirms Page.
3. If the Sub Amount (#of order * Average Price) on Trade_orders page is not equal to Sub Total.
4. If the Sub Amount on Trade_confirms page is not equal to Sub Total (#of order * Average Price).

ii. If Currency is same in Trade_Confirms file.

1. If the Total Amount charged does not matches Total Charge accounting for orders on Trade_Orders Page.
2. If Total Amount charged does not matches Total Charge for orders executed Trade_Confirms Page.
3. If the Sub Amount (#of order * Average Price) on Trade_orders page is not equal to Sub Total.
4. If the Sub Amount on Trade_confirms page is not equal to Sub Total (#of order * Average Price).
5. If the Total Amount charged matches Total Charge accounting for orders on Trade_Orders Page.
6. If Total Amount charged matches Total Charge for orders executed Trade_Confirms Page.
7. If the Sub Amount (#of order * Average Price) on Trade_orders page is equal to Sub Total.
8. If the Sub Amount on Trade_confirms page is equal to Sub Total (#of order * Average Price).

iii. If Sell/Buy type is not consistent in both files for the orders.

5. write_data
   Functionality: To write the data to the output file.