# Project 1 – Customer Order Report Application

(300 points)

You work as a developer for a national retail shipping company, and have been tasked with the creation of a report-generation tool. Your employer needs an application which they can use to view customer order summary information. The application needs to pull order data in from a database, assemble the data and generate a report containing the order information placed by each customer.

## Programming Tasks & Requirements

### Loading the Database Tables

Your program won't connect to a live database. You will instead load two data tables from a dataset member which will be provided for you. The .jcl skeleton accompanying this document will automatically take care of the member access for you, so you will only need to implement reading records as you've seen in lecture.

You will load each table in sequence from input. The first table contains the list of customers, including identifiers and shipping information. In addition, each customer has a preferred member status that is used to apply discounts to the order totals which your program will calculate. The second table contains order information for each customer, including transaction data such as order subtotal, payment method and the status of the order, which may be open or closed.

The table format specifications are given below under the Table Specifications section; use them to help determine how much storage you'll need to allocate per table. Before you begin writing the code for this assignment, you will want to write a small program to print the contents of the dataset member so you may inspect the records your program will need to parse.

### Processing the Tables

Your program will need to go through each customer in the customer information table, in order, and use its customer ID to retrieve the orders submitted by that customer. Each order has a base subtotal field that your program will use to calculate a final total by first subtracting a discount based on Preferred Member Status and then adding in the shipping cost. **Note:** We will not worry about sales tax.

$$\text{Total} = \text{Subtotal} - \text{Discount} + \text{Shipping}$$

The amount of the discount applied is determined by the membership status. The following table lists the types as well as the associated discount.

| Membership | Discount |
|---|---|
| None | 0 |
| BRONZE | 25 |
| SILVER | 50 |
| GOLD | 75 |

Your program will also need to calculate several values for each customer. Customers may have both open and closed orders; open orders are those for which payment has not yet been received, while closed orders have. Your program should count the numbers of both. It also needs tally the total amounts of open and closed orders separately. Finally, it will need to calculate the total number of orders and the total and average amounts for both open and closed orders.

## Generating the Report

Your program must generate a separate section for each customer in the customer information table. For each customer, include the customer's ID, the name of the customer, their shipping address and preferred member status. Your program should then list the details of each order, as well as the total for each order, followed by the summary statistics. The general reporting format for each customer is given below.

```
Customer: ...              Customer ID: ...               Membership: ...
Address: ...

Orders:
  ID   Status   Payment Method   Subtotal   Discount   Shipping   Total
  ========================================================================
  ...  ...      ...              ...        ...        ...        ...
  ========================================================================
                            Closed Orders: ...       Received Total: ...
                              Open Orders: ...        Pending Total: ...
                             Total Orders: ...        Overall Total: ...
                                                      Avg Order Amt: ...
```

## Requirements & Recommendations

You must use loops to read in and process the data tables. You do not need to use inner loops to process columns for either, however. You will instead want to write code to store or read from each column individually as needed. In addition, you must use `DSECTs` for column access into the rows of your tables, and so each table should get its own `DSECT`. Check the table format documentation for details on how the columns in your tables (and `DSECTs`) should be set up. **NOTE:** *The order of the column data in the records read from file are exactly the same as the order given in the table specifications.*

Note that as we don't have a good way to deal with decimal values yet, you'll simply perform integer arithmetic. You don't need to worry about prefixing '$' in front of dollar values or shortening the output fields for `XDECO`-converted values to less than 12 characters. We'll encounter a better way to handle this kind of tightening up of the output fields later than relying and working around `XDECO`.

You may want to use one or more `DSECTs` to help format the output buffer, especially when building the Orders table. There are several ways to handle formatting the report lines; then simplest is to simply have a separate buffer declared for each one. You may also elect to build output dynamically, but if you do so keep in mind that there will be several points at which you'll need to use destructive overlap to clear the contents of the output buffer, so consider carefully when it would make sense to use it.

## Table Specifications

Table 1: Customer information table

| Customer ID | Customer Name | Shipping Address | Preferred Member Status |
|---|---|---|---|
| F | CL20 | CL30 | CL6 |

Table 2: Order information table

| Customer ID | Order ID | Order Subtotal | Shipping Charge | Payment Method | Status |
|---|---|---|---|---|---|
| F | F | F | F | CL10 | CL6 |

## Coding and Documentation Standards

Assuring that your code follows proper coding conventions and documentation standards is a critical habit in the workplace, and can make the difference between keeping or losing your job. Your work will therefore be graded, in part, on your ability to adhere to the following coding and documentation standards:

### Coding standards

1. **Available registers.** You may only use registers 2 through 11 (inclusive) for loading values, addresses and performing math operations. Registers 0–1 and 12–15 are reserved for special use.
2. **DSECT usage.** Where specified, you must declare and use DSECTs. If the use of a DSECT is suggested, then you are encouraged but not required to use them.

### Documentation standards

1. **Documentation box.** Your program code must contain a standard documentation box (or docbox) immediately following the opening JCL statements:

```
*****************************************************************
* CSCI 360-2               PROJECT n                SPRING 2022 *
*                                                               *
* NAME: (Your name)                                             *
* DATE: (Date of submission)                                    *
*                                                               *
* (Brief Description)                                           *
*                                                               *
* REGISTER USAGE:                                               *
*   Ra - (Usage description)                                    *
*   Rb - (Usage description)                                    *
*   ...                                                         *
*                                                               *
*****************************************************************
```

$n$ – Number of the current lab exercise (i.e., 1)
$a$ – Lowest register number

$b$ – Next register number

2. **Inline documentation.** Roughly 80% of your lines of code should have comments separated from and following the operands of the Assembler instructions.

If your submitted code violates *any* of these standards, then you will automatically lose 50% of the points ***you would have received***, ***AFTER*** your program's functionality and output are assessed.