

/** Solution Q1***/

/*COMMENTS:

There are 3 character 2 numeric variables respectively*/

/**SAS codes Solution Q2***/

```
proc sort data=creditcard nodupkey out=creditcard_1;  
  by Cardtype Owner;  
run;
```

/*COMMENTS:

All the recorder with the same values in Cardtype and Owner are de-duplicated even though the values in other columns of de-duplicated records could be different

*/

/**SAS codes Solution Q3***/

```
proc sort data=creditcard noduprec out=creditcard_2;  
  by Cardtype Owner;  
run;
```

/*COMMENTS:

All the recorder with the same values on the whole row are de-duplicated. As you used 'noduprec' option, SAS will first check the values on Cardtype and Owner (They are treated as the primary key or identity) and then check other columns. If the values of all columns are identical then do de-duplication

*/

/**SAS codes Solution Q4***/

```
proc print data=creditcard_2;  
  Where Owner='Big Banks';  
  by Cardtype;  
run;
```

/*COMMENTS:

Note you should use 'where' rather than 'IF...ELSE' statement in SAS procedure. Make sure to sort data before you use 'BY' statement in SAS procedure. As you have sorted the data by 'Cartype' column in Q2 and Q3, you do not have to sort data again.

*/

/**SAS codes Solution Q5***/

```
proc print data=creditcard_2;  
  var Area Balance limit;  
  sum Balance;  
  by Cardtype;  
run;
```

/*COMMENTS:

As the output is displayed by each 'Cardtype' group, you should use 'BY Cardtype' statement. The variable 'Balance' is summed within each 'Cardtype' group, so you should specify the 'sum Balance' statement.

Reminder: You do not have to use 'SUMBY' statement here because by default the 'BY' statement is actually the 'SUMBY all variables in BY statement' statement. You will use the 'SUMBY' statement only if you want to sum by part of variables in 'by' variables list
*/

/**SAS codes Solution Q6***/

```
proc print data=creditcard_2;  
  var Area Balance limit;  
  sum Balance;  
  by Cardtype Owner;  
  sumby Cardtype;  
run;
```

/*COMMENTS:

In this problem the 'SUMBY' variable are not the same as those of 'BY' statement. Since you only want to sum by 'Cardtype' rather than both 'Cardtype' and 'Owner', you should use 'SUMBY' statement. Reminder: the 'SUMBY' variable must be the subset of the 'BY' variable list
*/

/**SAS codes Solution Q7***/

```
proc print data=callcenter(obs=5);  
run;
```

/*COMMENTS:

Use the 'OBS=n' option (adding bracket) following the data set name statement specified in 'PROC PRINT'
*/

/**SAS codes Solution Q8***/

```
proc sort data=callcenter;
  by BUSINESS_HOME;
run;
Proc transpose data=callcenter out=callcenter_tr;
  var last_hours;
  by BUSINESS_HOME;
  id week;
run;
```

/*COMMENTS:

Use 'BY BUSINESS_HOME' statement to transpose data. You should use the statement 'var last_hours' because you want to transpose the row values of only column 'last_hours' in the original table. Additionally, as you can see that the column names (i.e. MONDAY, TUESDAY,..) in the resulting table 'callcenter_tr' are actually the row values of column 'week' in original table, you should add 'id week' statement in 'PROC TRANSPOSE'

*/

/**SAS codes Solution Q9***/

```
Proc transpose data=product out=product_tr prefix=P name=item;
  var tran_count tran_amount;
  id product_id;
  idlabel product_type;
run;
```

/*COMMENTS:

As you can see in the resulting table, the displayed column names 'Food', 'Clothes', 'Toy', 'Jewelry' and 'Drink' are all labels (rather than real names), so you should add the 'IDLABEL' statement if you need to use the row values of column 'product_type' in the original table as the labels in resulting table.

The real column names in resulting table should come from the column 'product_id' (because you use 'id product_id' statement) and 'prefix=P' option, i.e. P1,P2,..P5

*/