```
%let path=/courses/d649d56dba27fe300/STA5067/SAS Data;
libname orion "&path/orion";
/*     1.a       */
%macro custtype(type);
   %let type=%upcase(&type);
    proc print data=orion.customer_dim;
       var Customer_Group Customer_Name Customer_Gender
           Customer_Age;
       where upcase(Customer_Group) contains "&type";
       title "&type Customers";
    run;
%mend custtype;
%custtype(Internet)
/*     1.b       */
%macro custtype(type) / minoperator;
%let type = %upcase(&type);
%if &type in GOLD  INTERNET %then %do;
    proc print data=orion.customer_dim;
       var Customer_Group Customer_Name Customer_Gender
           Customer_Age;
       where upcase(Customer_Group) contains "&type";
       title "&type Customers";
    run;
%end;
%else %do;
    %put ERROR: Value of TYPE: &type is not valid.;
    %put ERROR: Valid values are INTERNET or GOLD;
%end;
%mend custtype;
/*     1.c       */
options mlogic mprint;
%custtype(Internet)
%custtype(aaaa)
options nomlogic nomprint;
/*     1.d       */
%macro custtype(type) / minoperator;
%if &type= %then %do;
    %put ERROR: You must provide a value for TYPE;
    %put ERROR: Valid values are INTERNET or GOLD;
    %end;
%else %do;
   %let type=%upcase(&type) ;
   %if &type in GOLD  INTERNET %then %do;
    proc print data=orion.customer_dim;
       var Customer_Group Customer_Name Customer_Gender
           Customer_Age;
       where upcase(Customer_Group) contains "&type";
       title "&type Customers";
    run;
    %end;
    %else %do;
    %put ERROR: Value of TYPE: &type is not valid.;
```

```
    %put ERROR: Valid values are INTERNET or GOLD;
    %end;
%end;
%mend custtype;
/*     1.e       */
options mlogic mprint;
%custtype(Internet)
%custtype(aaaa)
%custtype()
options nomlogic nomprint;
/*     2.a       */
%macro listing(custtype);
   proc print data=orion.customer noobs;
   run;
%mend listing;
%listing(2010)
/*     2.b       */
%macro listing(custtype);
%if &custtype= %then %do;
   proc print data=orion.customer noobs;
   var Customer_ID Customer_Name Customer_Type_ID;
   title "A Listing of All Customers";
   run;
%end;
%else %do;
   proc print data=orion.customer noobs;
   where Customer_Type_ID =&custtype;
   var Customer_ID Customer_Name;
   title "A Listing of &custtype Customers";
   %end;

%mend listing;
/*     2.c       */
options mlogic mprint;
%listing(2010)
%listing()
options nomlogic noprint;
/*     2.d       */
%macro listing(custtype) / minoperator;
proc sql noprint;
select distinct Customer_Type_ID into :IDlist separated by ' '
from orion.customer_type
;
quit;
%put &IDlist;
%if &custtype= %then %let Flag=0;
%else %if &custtype in &IDlist %then  %let Flag = 0;
      %else  %let Flag = 1;
%if &Flag=0 %then %do;
   %if &custtype= %then %do;
   proc print data=orion.customer noobs;
   var Customer_ID Customer_Name Customer_Type_ID;
   title "A Listing of All Customers";
```

```
    run;
    %end;
    %else %do;
    proc print data=orion.customer noobs;
    where Customer_Type_ID =&custtype;
    var Customer_ID Customer_Name;
    title "A Listing of &custtype Customers";
    %end;
%end;
%else %do;
%put ERROR:Value for CUSTTYPE is invalid.;
%put ERROR:Valid values are 1010 1020 1030 1040 2010 2020 2030 3010 ;
%end;
%mend listing;
/*     2.e       */
options mlogic mprint;
%listing()
%listing(2010)
%listing(20)
options nomlogic nomprint;

/*     3.a       */
%macro generatecode(bartype=VBAR, dims=3D,
                    var=Customer_Age_Group, color=pink,
                    surface=S);
    proc gchart data=orion.customer_dim;
       &bartype&dims &var;
       pattern color=&color value=&surface;
    run;
    quit;
%mend generatecode;

%generatecode();

/*     3.b       */
%macro generatecode(bartype=VBAR, dims=3D,
                    var=Customer_Age_Group, color=pink,
                    surface=S) /minoperator;
%let total1= vbar hbar 3D null s x1 x2 x3 x4 x5;
%if not(&bartype in &total1) %then %do;
%let m1=0;
%end;
%else %let m1=1;
%if not(&dims in &total1) %then %do;
%let m2=0;
%end;
%else %let m2=1;
%if not(&surface in &total1) %then %do;
%let m3=0;
%end;
%else %let m3=1;
%let m=%eval(3-&m1-&m2-&m3);
```

```
    %if &m=0 %then %do;
    proc gchart data=orion.customer_dim;
       &bartype&dims &var;
       pattern color=&color value=&surface;
    run;
    quit;
    %end;
    %else %do;
    %put ERROR: Invalid Bar Type was supplied.Valid Values are VBAR or HBAR.;
    %put ERROR: Invalid Dimension Value.The value can be 3D or a null value.;
    %put ERROR: Invalid Surface Value.The value can be S, X1, X2, X3, X4, X5.;
    %put ERROR: Due to parameter errors SAS code will not execute.;
    %put You have &m errors;
    %put NOTE: There were 77 observations read from the data set ORION.CUSTOMER_DIM.
         ;
    %end;
%mend;

options mlogic mprint;
%generatecode(bartype=sbar, dims=1t ,surface=99)
%generatecode(bartype=sbar, dims=3D ,surface=99)
%generatecode(bartype=vbar, dims=3D ,surface=x1)
options nomlogic nomprint;

/*     4.a       */
proc means data=orion.order_fact sum mean maxdec=2;
   where Order_Type = 2;
   var Total_Retail_Price CostPrice_Per_Unit;
   title "Summary Report for Order Type 2";
run;

/*     4.b       */
%macro style;
%do i = 1 %to 3;
    proc means data=orion.order_fact sum mean maxdec=2;
   where Order_Type = &i;
   var Total_Retail_Price CostPrice_Per_Unit;
   title "Summary Report for Order Type &i";
   run;
%end;
%mend style;

options mlogic mprint;
%style
options nomlogic nomprint;

/*     5.a       */
%macro tops(obs=3);
    proc means data=orion.order_fact sum nway noprint;
       var Total_Retail_Price;
       class Customer_ID;
       output out=customer_freq sum=sum;
    run;
```

```
    proc sort data=customer_freq;
       by descending sum;
    run;

    data _null_;
       set customer_freq(obs=&obs);
       call symputx('top'||left(_n_), Customer_ID);
    run;
%mend tops;

%tops()
%tops(obs=5)

/*     5.b       */
%macro tops(obs=3) ;
    proc means data=orion.order_fact sum nway noprint ;
       var Total_Retail_Price;
       class Customer_ID;
       output out=customer_freq sum=sum;
    run;

    proc sort data=customer_freq;
       by descending sum;
    run;

    data _null_;
    set customer_freq(obs=&obs) end=final;
    length top $30.;
    retain top;
    top=catx(',',top,trim(Customer_ID));
    if final then call symputx('Topx',top);
    run;

    proc print data= orion.customer_dim;
    var Customer_ID Customer_Name Customer_Type;
    where Customer_ID in (&topx);
    title "top&obs Customers";
    run;
%mend tops;

options mlogic mprint;
%tops()
%tops(obs=5)
opions nomlogic nomprint;

/*      6.a        */
%macro memberlist(custtype);
   proc print data=Orion.Customer_dim;
      var Customer_Name Customer_ID Customer_Age_Group;
      where Customer_Type="&custtype";
      title "A List of &custtype";
   run;
```

```
%mend memberlist;

%macro listall;
   data _null_;
      set orion.customer_type end=final;
      call symputx('type'||left(_n_), Customer_Type);
      if final then call symputx('n',_n_);
   run;
   %put _user_;
%mend listall;

%listall

/*     6.b       */
%macro memberlist(custtype);
   proc print data=Orion.Customer_dim;
      var Customer_Name Customer_ID Customer_Age_Group;
      where Customer_Type="&custtype";
      title "A List of &custtype";
   run;
%mend memberlist;

%macro listall;
   data _null_;
      set orion.customer_type end=final;
      call symputx('type'||left(_n_), Customer_Type);
      if final then call symputx('n',_n_);
   run;
%do i=1 %to &n;
%memberlist(&&type&i)
%end;
%mend listall;
%listall

/*      7.a        */
%macro varscope;
   data _null_;
      set orion.customer_type end=final;
      call symputx('localtype'||left(_n_), Customer_Type);
      if final then call symputx('localn',_n_);
   run;
   %put _user_;
%mend varscope;
%varscope
/*      7.b      */
%macro varscope;
   proc sql noprint;
   select distinct count(Customer_Type) into :m
   from orion.customer_type;
   quit;

   %do i=1 %to &m;
   %local localtype&i;
```

```
    %end;
    %local localn;

    data _null_;
       set orion.customer_type end=final;
       call symputx('localtype'||left(_n_), Customer_Type);
       if final then call symputx('localn',_n_);
    run;

%put _user_;
%mend varscope;
%varscope
```