|  |
| --- |
| **Name: Ayush Patel**  **Roll: A232**  **Semester: VII**  **Class: A**  **Branch: Btech IT (4th year)** |
| **Practical-6 Part A** |
| **Aim:** To create a SAS data set by reading data from a raw data file.  **A**     1. Write a SAS Program to read SAS raw Data File sales.csv and create output dataset. 2. Write Proc print to display output dataset. 3. For above dataset create apply SAS formats for salary variables. 4. Create user defined format for gender and country to replace gender code and country code with meaningful name. 5. Create permanent format birthdate as date9.   B   1. Create Bonus variable 10 percent of salary; 2. Use informats to read non standard data for above sales.csv file and use keep and drop statement and use sub setting if statement to include records having bonus greater than 500. 3. Create SAS dataset Phone by reading Phone2.csv file . Phone.csv as shown below        1. Display report Phone with formats, label, title and footnote statement and use keep and drop statement . |
| **Prerequisite:** Programming for Problem Solving |
| **Outcome:** To Reading and Writing Raw data files. |
| Theory:Raw Data FilesA raw data file is also known as a *flat file*.They are text files that contain one record per line.A record typically contains multiple fields.Flat files do not have internal metadata.External documentation, known as a *record layout*, should exist.A record layout describes the fields and locations within each record.Fields in Raw Data FilesIn order for SAS to read a raw data file, you must specify the following information about each field:the location of the data value in the recordthe name of the SAS variable in which to store  the datathe type of the SAS variable List InputUse list input to read delimited raw data files.SAS considers a space (blank) to be the default delimiter.Both standard and nonstandard data can be read.Fields must be read sequentially, left to right.Reading a Delimited Raw Data FileUse *INFILE* and *INPUT* *statements* in a DATA step to read a raw data file.Syntax:SAS Code Example |
| **Procedure:**   1. Open SAS Studio and write the SAS program |
| **Instructions:**   1. Write source code of all stored procedure 2. Copy code & paste in code section of Part B. |
| **Part B** |
| **Code:**  /\*A\*/  /\*1\*/  data datasubset1;  infile '/home/u59005730/sasuser.v94/sales.csv' dlm= ',';  input Emp\_Id Fname $ Lname $ Gender $ Salary Job\_Title $ Country $ BirthDate:date9. HireDate:ddmmyy10.;  run;  /\*2\*/  proc print data = datasubset1;  run;  /\*3\*/  proc print data = datasubset1;  format Salary dollar10.2;  run;  /\*4\*/  proc format;  value $ctryfmt 'AU' = 'Australia'  'US' = 'USA';  value $genfrmt 'F' = 'Female'  'M' = 'Male';  run;  proc print data = datasubset1;  format Salary dollar10.2;  format Country $ctryfmt. Gender $genfrmt.;  run;  /\*5\*/  data work.datasubset2;  set work.datasubset1;  format BirthDate date9.;  format Salary dollar10.2;  format Country $ctryfmt. Gender $genfrmt.;  run;  proc print data = work.datasubset2;  run;  proc contents data = work.datasubset2;  run;  /\*B\*/  /\*a\*/  data work.datasubset3;  set work.datasubset2;  Bonus = Salary \* .10;  run;  proc print data = work.datasubset3;  run;  /\*b\*/  data work.datasubset4;  set work.datasubset3;  keep Emp\_Id Fname Salary Bonus;  where Bonus > 500;  run;  proc print data = work.datasubset4;  run;  data work.datasubset4;  set work.datasubset3;  drop Lname Gender Country;  where Bonus > 500;  run;  proc print data = work.datasubset4;  run;  /\*c\*/  data dataPhone;  informat name $40. home $20. office $20.;  infile '/home/u59005730/sasuser.v94/phone2.csv' dlm= ',' dsd;  input name $ office $ home $;  run;  proc print data = work.dataPhone;  run;  /\*d\*/  proc print data = dataPhone label;  var name office home;  title "Phone Directory";  label name = 'Name'  office = 'Office'  home = 'Home';  footnote "Confidential Data"; |
| **Output:**  **A)**  **1)**    **2)**    **3)**    **4)**    **5)**    **B)**  **a)**    **b)**    **c)**    **d)** |
| **Observation & Learning:**  Learnt to create a SAS data set by reading data from a raw data file in SAS Studio. |
| **Conclusion:**  Successfully created a SAS data set by reading data from a raw data file in SAS Studio and formatting it. |
| **Question:**  **Q1. What is SAS Format? Explain Syntax of SAS Format.**  **Ans:**  SAS format is the instruction that specifies how the value of a variable should be printed or displayed and SAS informats are the specification for how raw data should be read.  Syntax of SAS format:  **<$>format<w>.<d>**  where : $ -> indicates a character format.  format -> Names the SAS format.  w -> specifies the total format width, including decimal places and special characters.  . is required syntax. Formats always contain a period(.) as part of the name.  d -> specifies the number of decimal places to display in numeric formats.  **Q2. Explain difference between SAS format and user defined formats.**  **Ans:**  SAS built in format are used to make the data appear in the predefined manner but using user defined formats the data appearance can be customised according to users need.  **Q3. Explain different types of SAS inbuilt formats with example.**    **Screenshot 2021-08-05 at 11.32.10 PM.png** |