Richland College
DALLAS COUNTY COMMUNITY COLLEGE DISTRICT
School of Engineering and Technology

COSC1436 – PROJECT

Title	File Access – Sale report	
Time to complete	4 weeks	
COURSE OBJECTIVES – LEARNING OUTCOME		LAB OBJECTIVES
[LO1] Complete each program through all steps of the programming process: Analyze the problem requirement, provide the pseudo-code or flowchart of the design, write the C++ code, compile, debug and run the program [LO2] Apply the technique of top-down design. Provide the design of the algorithm from the requirement statement into a structured program by writing the pseudo-code or the flow-chart [LO3] -Declare variables of data types, such as int, bool, char, float, double, string; assign values, represent, access, manipulate and store them into computer as local or as global variables [LO4] -read input from the keyboard and from the input file -display output to the screen and to the output file -open and close input files and output file [LO5] Use if, if..else, nested if or switch statement in programs [LO6] Use for, do..while and while loop in programs [LO7] -Access predefined function -Create and use value-returning functions or void functions [LO8] -Use value and reference parameters in void function and returning functions [LO10] Enhance the problem solving skills		After finish the lab3 students can do the following: -Complete the lab on time (Time Management) -Ensure doing the lab by following the program process: Write the pseudo-code, Create a new project, add source file to the project, compile and run the program without errors and qualified to the requirement -Declare variables -read from the keyboard by using cin, cin.get and getline -access variables after initializing -Calculate sum, subtraction, multiplication, division with variables -Print the output in columns by using setw(n), left, right -Format the decimal numbers with 2 digits after decimal point by using showpoint, setprecision(2), fixed -Can use if, if..else or switch statement at correct situation -Can use do.. while loop to manage the menu to re-display -Can define and access user-defined functions -can open file to read and write -can access predefined functions such as, substr, etc.

<table>
<tr>
<td>Skills required to do the Lab 4</td>
<td>To do this lab, students have to know:
(If you want to see example how to do these, read topics at QA FOR LABS on eCampus)
-What is pseudo-code?
-How to write comment lines in the program by using “//” at the beginning of the line
-Declare variable of data types: float, String
-Read values from the keyboard use: cin, cin.get, getline
-Read a string including space from the keyboard use: getline(cin, stringVariable);
-Assign a values to a variables
-Use Math calculation on int, float variables.
-Display output on the screen
-How to display the numbers with 2 decimal digits and straight in one column: use set:
cout << fixed << showpoint;
cout << setprecision(2) << fixed;
-How to pause the output screen for reading
-how to write the if, if..else and switch statement
-how to generate the current date
char currentDate[10];
_strdate_s(currentDate);
string date(currentDate);
cout << "date: " << date;

-how to use substr:
mm = date.substr(0, 2);
dd = date.substr(3, 2);
yy = date.substr(6, 2);

-How to display information in columns:
cout << fixed << showpoint << setprecision(2);
cout << setw(13) << left << "Balance: "<< setw(32) << right << paid - total << "\n"</td>
</tr>
<tr>
<td>HOW TO DO</td>
<td>*Step1: Create the pseudo-code → store it to file pseudoCode_yourLastName
*Step2: Write the code
-start Virtual Studio C++, create the project → provide the project name
-add .cpp file → provide the application name
-follow the pseudo-code and use C++ to write the code of the program
*Step3: compile and run the program
*Step4: debug if there is any errors to complete the program</td>
</tr>
</table>

PROJECT	**REQUIREMENT** (change yourLastName to your own last name) The company sales several types of products: SU6191 - price 10.99 per unit, SU6192– price 12.99 per unit, SU6193 – price 14.99 per unit. The company ask for the application to allow their employees to calculate and print out the receipt on the sreen for each sale transaction. Also, the application allow users can print out the sale report at the end of a day, at the end of month and at the end of a year To do that, create the project named as **SU2019_PROJECT_YourLastName** the add then application named as **SaleSU619Report_yourLastName** The application SaleSU19Report_yourName, first, display the menu (CHANGE Iames Smith to your name) SALE SUMMER 2019 PRODUCTS – James Smith Today: 06/15/19 1. Sale Product 2. Ending day sale report 3. Ending month sale report 4. Ending year sale report 0. Exit Read the task After finishing one task, you should re-display the menu to allow users to select other tasks until users want to exit **TASK1: SALE PRODUCT** For each sale transaction: -Display the message to ask and read the input from the keyboard for the following information: • How many units of Model U6191 • How many units of Model SU6192 • How many units of Model SU6193 • The amount paid -Apply the formula to calculate the charge then display the receipt on the screen Sale model SU6191 = unit price of SU6191 * number of unit model SU6191 sold Sale model SU6192 = unit price of SU6192 * number of unit model SU6192 sold Sale model SU6193 = unit price of SU6193 * number of unit model SU6193 sold Subtotal = Sale model SU6191 + Sale model SU6192 + Sale model SU6193 Tax = subtotal * 8.25% Total = subtotal + Tax Balance = amount paid – total

The output of the receipt should be (the numbers should be aligned to the right and have 2 decimal digits)
The transaction is start dd0001 at the first sale transaction during the day where dd is the date 15 in 06/15/19 then increase by 1 in each next transaction

```
.........................................
        RECEIPT - SALE SU0619 PRODUCT
.........................................
Date:                                 06/15/19
Sale transaction:                       150001
.........................................
Model SU6191 (10.99/per unit)     10    109.90
Model SU6192 (12.99/per unit)     10    129.90
Model SU6193 (14.99/per unit)     10    149.90
.........................................
Sub total:                              389.70
Tax(8.25%)                               32.15
Total:                                  421.85
Amound paid:                            500.00
Balance:                                 78.15
```

Also, the sale transaction should be written to the output file named daySale_yymmdd.txt
For example **daySale_190615.txt** where:
-The first column is transaction number
-The second column is number of product SU6191 units sold
-The third column is number of product SU6192 units sold
-The last column is number of product SU6193 units sold

```
150001    10    10    10
150002    12    12    12
150003    14    21    18
150004    12    18    10
150005    10     8    12
150006    15    12    14
150007    15    15    15
150008    20    12    12
150009     8    15    10
150010    10     8    12
```

TASK 2: ENDING DAY REPORT
At the end of one day, the users will run the day report
Select the task 2 from the main menu
-Ask users for the date to run the report. The input should be in the format: mm/dd/yy
-split the input into mm, dd, yy then create the file name **daySale_yymmdd.txt**
-open file daySale_yymmdd.txt (for example daySale_190615.txt to read

-for each line, split to get: number of SU6191 units, number of SU6192 units, number of SU6193 units ➔ keep adding up total of SU6191 units, total of SU6192 units, total of SU6193 units
-After the last transaction, close file then calculate and display on the screen the day report:

SALE SU619 PRODUCTS – 06/15/19

Model SU6191 (10.99/per unit)	126	1384.74
Model SU6192 (12.99/per unit)	131	1701.69
Model SU6193 (14.99/per unit)	125	1873.75
…….		
Sub total:		4960.18
Tax(8.25%)		409.21
Total:		5369.39

Also, open the output file named monthSale_yymm.txt, for example, monthSale_1906.txt
And write one line to the file then close file
For example: in the file **monthSale_1906.txt** where the first column is day (from 1 to 30 for June and 15 for 06/15/19), second column is number of SU6191 units sold, the third column is number of SU6192 units sold, the last column is the number of SU6193 units sold (see whole file in the file monthSale_1906.txt downloaded from eCampus)

15 126 131 125

TASK 3: ENDING MONTH REPORT
At the end of month, the users will run the month report
Select the task 3 from the main menu
-Ask users for the month to run the report. The input should be in the format: mm/yy
-split the input into mm, yy then create the file name **monthSale_yymm.txt**
-open file monthSale_yymm.txt (for example monthSale_1906 .txt to read (download file from eCampus)
-for each line, split to get: number of SU6191 units, number of SU6192 units, number of SU6193 units ➔ keep adding up total of SU6191 units, total of SU6192 units, total of SU6193 units
-After the last line, close file then calculate and display on the screen the month report:

SALE SU619 PRODUCTS – 06/19

Model SU6191 (10.99/per unit)	3870	42531.30
Model SU6192 (12.99/per unit)	3976	51648.24
Model SU6193 (14.99/per unit)	3849	57696.51
…….		
Sub total:		151876.05
Tax(8.25%)		12529.77
Total		164405.82

Also, open the output file named yearSale_yy.txt, for example, yearSale_19.txt

	And write one line to the file then close file For example: in the file **yearSale_19.txt** where the first column is month (from 1 to 12, 3 for March), second column is number of SU6191 units sold, the third column is number of SU6192 units sold, the last column is the number of SU6193 units sold (see whole file in the file yearSale_19.txt downloaded from eCampus) 06 3870 3976 3849 **TASK4: ENDING YEAR REPORT** At the end of year, the users will run the year report Select the task 4 from the main menu -Ask users for the year to run the report. The input should be in the format: yyyy -get the last two digits yy then create the file name **yearSale_yy.txt** -open file yearSale_yy.txt (for example yearSale_19.txt to read (download file from eCampus) for each line, split to get: number of SU6191 units, number of SU6192 units, number of SU6193 units ➔ keep adding up total of SU6191 units, total of SU6192 units, total of SU6193 units -After the last line, close file then calculate and display on the screen the year report: SALE SU619 PRODUCTS – 2019 Model SU6191 (10.99/per unit) 38757 425939.43 Model SU6192 (12.99/per unit) 41686 541501.14 Model SU6193 (14.99/per unit) 40235 603122.65 …….. Sub total: 1570563.22 Tax(8.25%) 129571.47 Total 1699685.69

	<table><tr><td>turn in on time</td><td>8</td></tr><tr><td>Submit all files requested with correct names</td><td>3</td></tr><tr><td>compile success with all the requirements</td><td>10</td></tr><tr><td>Pseudo-code or flowchart</td><td>5</td></tr><tr><td>Write the comment in the program</td><td>5</td></tr><tr><td>Using user-defined functions and call them in main()</td><td>5</td></tr><tr><td>Manage main menu to re-display after finishing each task and terminate when users selects exit</td><td>2</td></tr><tr><td>TASK1 sale product: read input, calculate, print the receipt, write to file, close files</td><td>12</td></tr><tr><td>TASK2 Day Sale report: read file, calculate, print the moth report, write to file, close files</td><td>8</td></tr><tr><td>TASK3 Month Sale report: read file, calculate, print the report, wrie to file, close files</td><td>8</td></tr><tr><td>TASK4 YearSale report: read file, calculate, print the report, close file</td><td>6</td></tr><tr><td>Output as requested</td><td>8</td></tr><tr><td>PROJECT SCORES</td><td>80 POINTS</td></tr></table>
HOW TO TURN IN	You should turn in the following files: SU2019_PROJECT_YourLastName.exe SaleSU619Report_yourLastName.cpp Pseudo_code IF YOU GET ANY PROBLEM TO SUBMIT FILE .class, YOU CAN SUBMIT ALL PROJECT INTO ONE FILE .zip or .rar TO SEND