# 

**Multimedia University**

**CCP6114 Programming Fundamentals 2430**

**Lecture section: TC2L**

**Tutorial section: TT4L**

**Group number: ?04**

**Group leader student name: Feqhah Delilah Binti Mohd Faizul**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Student**  **ID** | **Student Name by**  **alphabetical order** | **Task Descriptions** | **Percentage (%)** |
| 1 | 242UC244FY | FEQHAH DELILAH BINTI MOHD FAIZUL |  |  |
| 2 | 242UC244CK | WAN HANANI IMAN BINTI AZIDI@SAPAWI |  |  |
| 3 | 242UC244L8 | VIDHYA DARINEY A/P RAJASINGAM |  |  |

Every student is responsible for 100% (task percentage) of this group assignment work.

# **Mark sheet checklist (30%)**

**Assignment programming and documentation (30%)**

You are required to submit assignment milestone 1 to your respective tutor also before the submission deadline.

Also document all your assignment tasks with this marking table that contain cover page, table of contents, page numbering, inputs, outputs, screenshots, explanations, and others.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Max** | **A1** | **A2** | **Mark** |
| Q1.  Create database and view database name  Create table, view table name  Table supports two data types i.e. INT, TEXT  Insert rows to the table  View table in csv mode | 5 | \* | \* | ? |
| Q2.  Reading from a file, outputting to screen, writing to a file (0 if no files used or no screen outputs) | 3 |  | \* | ? |
| Q3.  Update table rows and view table  Delete table rows and view table | 4 |  | \* | ? |
| Q4.  Count and output number of rows in the table | 2 |  | \* | ? |
| Q5.  Must use vectors or arrays, functions or classes, to store file output contents | 2 |  | \* | ? |
| Q6.  Inline comments, function or class comments, indentation, following proper C++ naming and styling conventions  Any violation is penalized by a reduction of 1 mark. | 2 |  | \* | ? |
| Q7.  The program demonstrates error handlings.  [0: Below Expectation, 1: Within Expectation, 2: Exceed Expectation] | 2 |  | \* | ? |
| Q8.  Correct structured diagrams | 2 |  | \* | ? |
| Q9.  Correct flowcharts or pseudocodes with explanations for all the file input statements.  Any missing flowchart or pseudocode will cause you to lose 1 mark. | 2 |  | \* | ? |
| Q10.  Sample file inputs at least 3, their screen outputs, their file outputs with screenshots and explanations. | 3 |  | \* | ? |
| Q11.  User documentation done and is coherence with the all implementations.  Any missing input statement will cause you to lose 1 mark. | 3 |  | \* | ? |
| Total | 30 |  |  | ? |

Additional comments

|  |
| --- |
|  |

You are required to fill in your task percentage and task descriptions.  
 Every student is responsible for 100% (task percentage) of this group assignment work.

Student 1

|  |  |
| --- | --- |
| Student ID | 242UC244FY |
| Student name | FEQHAH DELILAH BINTI MOHD FAIZUL |
| Task percentage |  |
| Task descriptions |  |
| Total score (30m) |  |

Student 2

|  |  |
| --- | --- |
| Student ID | 242UC244CK |
| Student name | WAN HANANI IMAN BINTI WAN MOHD AZIDI@SAPAWI |
| Task percentage |  |
| Task descriptions |  |
| Total score (30m) |  |

Student 3

|  |  |
| --- | --- |
| Student ID | 242UC244L8 |
| Student name | VIDHYA DARINEY A/P RAJASINGAM |
| Task percentage |  |
| Task descriptions |  |
| Total score (30m) |  |

Each feature will be evaluated based on documentation, fulfilment of requirements, correctness, compilation without warnings and errors, error free during runtime, error handlings, quality of comments, user friendliness, good coding format and style.

# 

# **Table of contents with page numbers and links**

[Cover page](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561243)

[Mark sheet checklist (30%)](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561244)

[Table of contents with page numbers and links](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561245)

[Delete this information section](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561246)

[Question Section](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561247)

[Q01, Q09, Q11 [5] Database name, table name, table of two data types, insert table rows, view table in csv mode](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561248)

[Q02, Q09, Q11 [3] Reading from a file, outputting to screen, writing to a file](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561249)

[Q03, Q09, Q11 [4] Update table rows, delete table rows, view table](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561250)

[Q04, Q09, Q11 [2] Count and output number of rows in the table](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561251)

[Q05, Q11 [2] Must use vectors or arrays, functions or classes, to store file output contents](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561252)

[Q06, Q11 [2] Inline comments, function or class comments, indentation, proper C++ naming with styling conventions](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561253)

[Q07, Q09, Q11 [2] The program demonstrates error handlings](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561254)

[Q08, Q11 [2] Structured diagrams](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561255)

[Q10, Q11 [3] Three sample input files, step by step screenshot outputs, output files, explanations](https://mmuedumy-my.sharepoint.com/personal/feqhah_delilah_mohd_student_mmu_edu_my/Documents/assignment%20que%20ccp6114%202430%20report%20template%20dynamic%20(Repaired).docx#_Toc184561256)

# **Question Section**

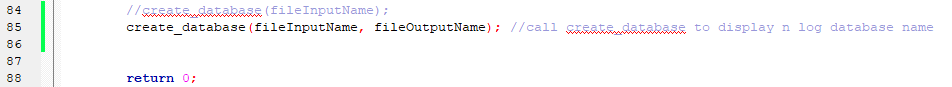
# **Q01, Q09, Q11 [5] Database name, table name, table of two data types, insert table rows, view table in csv mode**

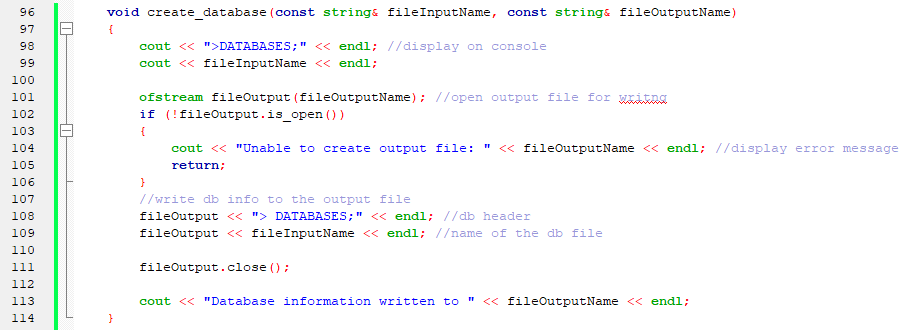
1. Create database and view database name
2. Create table, view table name
3. Table supports two data types i.e. INT, TEXT
4. Insert rows to the table
5. View table in csv mode

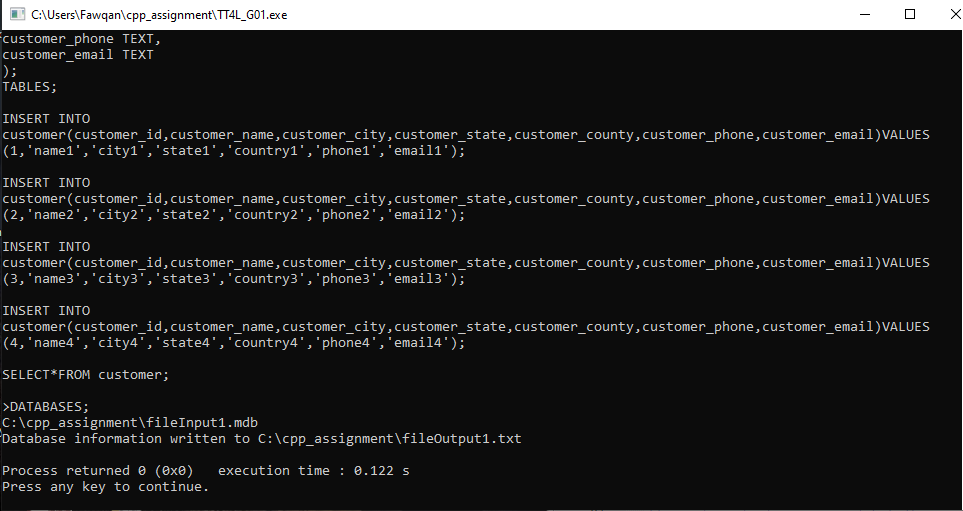
Screenshots (inputs, outputs), explanations

1. **Create database and view database name**

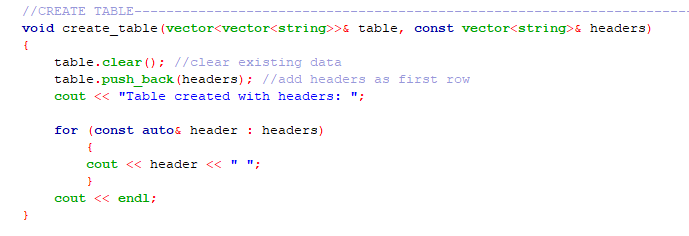
* Explanation : This function is responsible for displaying and recording the database name. First, it prints a header (> DATABASES;) and the database file name on the console for the user to see. It then saves this information into an output file. If the output file can't be created, the program notifies the user with an error message. Once the process is successful, the function confirms that the database name has been saved and provides feedback, ensuring the user is informed at every step





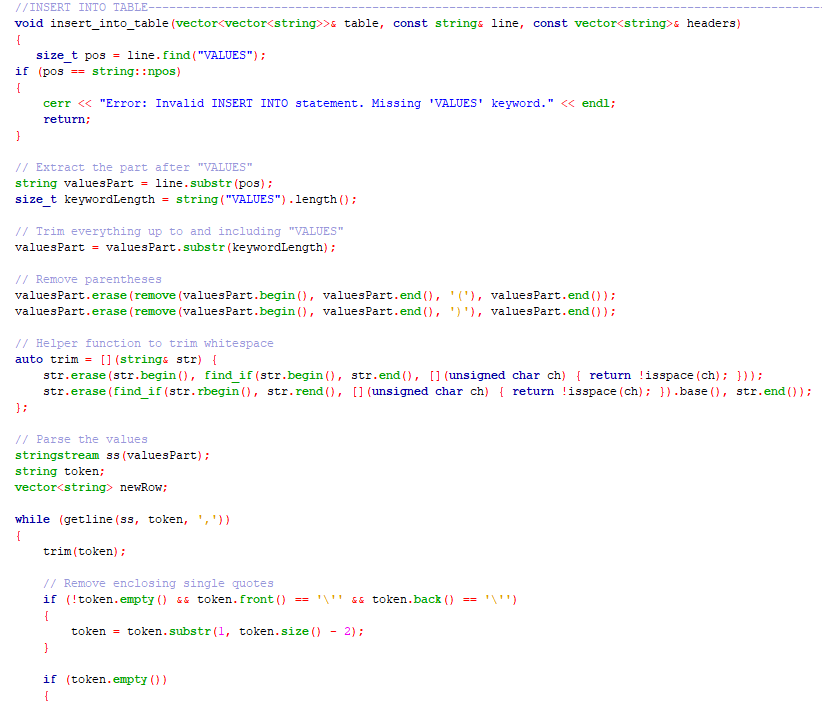


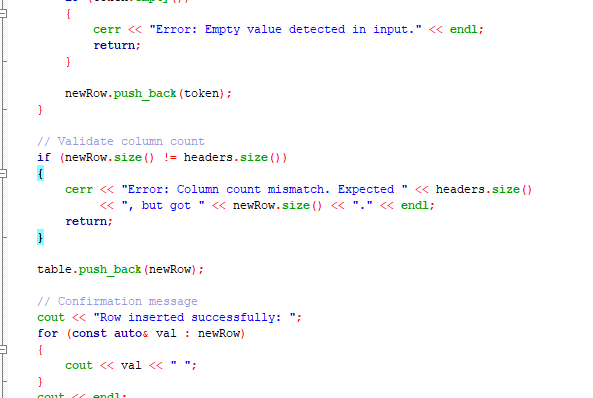
1. **Create table, view table**
2. **Table supports two data types i.e. INT, TEXT**

****

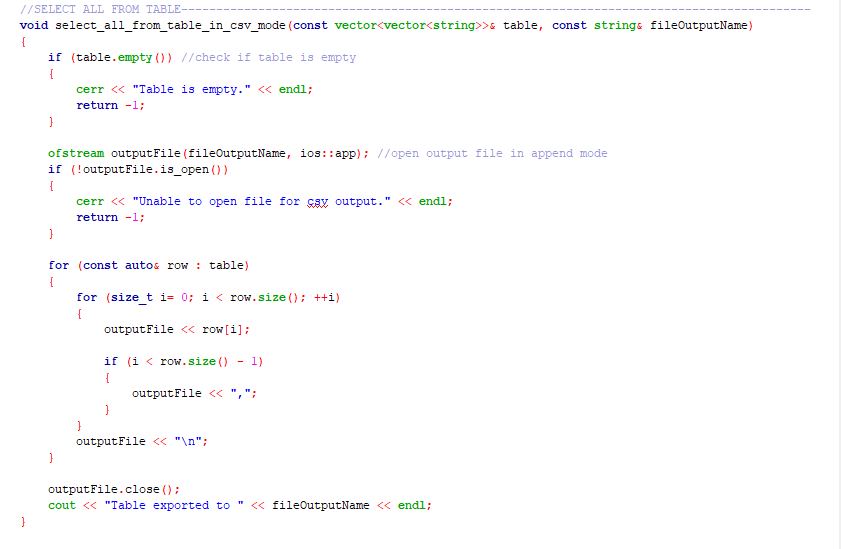
Each column is assigned a type, with the first being integer, and the rest text. As seen in the 2nd picture, when a non digit character is placed into the customer\_id column, which is an integer column, an error message is displayed.

1. **Insert rows to the table**

****

****

1. **View table in csv mode**

****

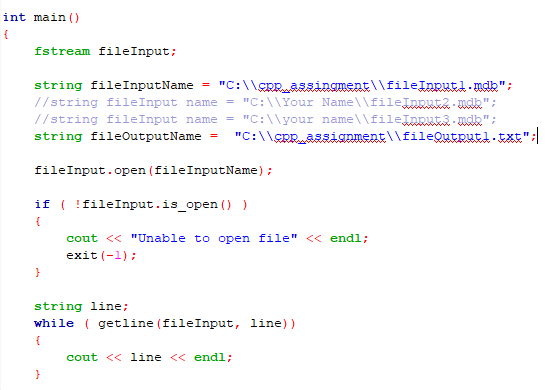
Pseudocode parts, explanations

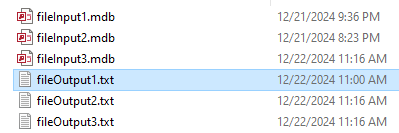
|  |
| --- |
| START  1. DEFINE file paths:  - INPUT\_FILE = "C:\\cpp\_assignment\\fileInput1.mdb"  - OUTPUT\_FILE = "C:\\cpp\_assignment\\fileOutput1.txt"  2. OPEN input file (INPUT\_FILE) for reading:  IF file cannot be opened:  DISPLAY "Unable to open file"  EXIT program  3. INITIALIZE empty TABLE structure:  - TABLE = [] (A 2d vector to hold table rows)  - tableName = “”  4.PROCESS Input\_File :  WHILE reading lines from INPUT\_FILE::  READ line  DISPLAY “Line read: [line content]” (debugging output)  IF the line contains “CREATE TABLE”:  DISPLAY “Creating table : “+ tableName  CALL create\_table(TABLE, tableName) function  ELSE IF the line contains “INSERT INTO”:  DISPLAY “Inserting data into table: “+ tableName  CALL insert\_into\_table() funct  6. CLOSE Input\_File :  7. CALL create\_database(INPUT\_FILE, OUTPUT\_FILE) to handle db information:  DISPLAY “>DATABASES;”  DISPLAY database name (INPUT\_FILE)  WRITE database information into OUTPUT\_FILE  8. EXPORT TABLE to csv format:  CALL select\_all\_from\_table\_in\_csv\_mode(TABLE, OUTPUT\_FILE)  DISPLAY “ Table exported to [csv\_file]  END  // —--------------------------------------------------------------------------------------------------//  // Function pseudocode //  Function : create\_database(INPUT\_FILE,OUTPUT\_FILE)  1. OPEN OUTPUT\_FILE for writing :  IF OUTPUT\_FILE cannot be opened  DISPLAY “Error: unable to create file: [OUTPUT\_FILE]”  RETURN  2. WRITE database information:  WRITE “>DATABASES;” to OUTPUT\_FILE.  WRITE name of the database (INPUT\_FILE) to OUTPUT\_FILE.  3.CLOSE OUTPUT\_FILE  4.DISPLAY “Database information written to [OUTPUT\_FILE] “  Function : create\_table(vector<vector<string>>& TABLE, const vector<string>& HEADERS)  1. CLEAR any existing data or table  2. OUTPUT HEADERS followed by a space  3. DISPLAY “Table created with headers“  4. DISPLAY HEADER  5. PRINT an empty line  Function : insert\_into\_table(vector<vector<string>>& TABLE, const string& line, const vector<string>& HEADERS)  1. FIND “values” in fileinput  DISPLAY “"Error: Invalid INSERT INTO statement. Missing 'VALUES' keyword."  RETURN  2. EXTRACT string values from fileinput  3. DELETE ‘(‘ ‘)’ from extracted string  4. DEFINE trim function  5. FOR every token, ADD ‘,’  DELETE whitespace  DELETE quotes ‘ ‘ ‘  6. IF size newRow is not equal to size headers  PRINT "Mismatch in number of columns."  EXIT  7. APPEND newRow to table  8. DISPLAY "Row inserted successfully."  Function : select\_all\_from\_table\_in\_csv\_mode(const vector<vector<string>>& TABLE, const string& fileOutputName)  1. CHECK if table has content  IF table is empty  DISPLAY "Table is empty."  EXIT program  2. OPEN OUTPUT\_FILE in append mode  IF OUTPUT\_FILE is not open  DISPLAY "Unable to open file for csv output."  EXIT program  3. FOR each row in table  FOR every element in row  WRITE element in OUTPUT\_FILE  IF element is not last element  WRITE "," in OUTPUT\_FILE  WRITE new line in OUTPUT\_FILE  4. CLOSE OUTPUT\_FILE  5. DISPLAY “Table exported to [OUTPUT\_FILE] “ |

# **Q02, Q09, Q11 [3] Reading from a file, outputting to screen, writing to a file**

\*0 if no files used or no screen outputs\*

Screenshots (inputs, outputs), explanations

1. Added fileInput1.mdb & fileOutput1.txt in the folder with all the data input & output

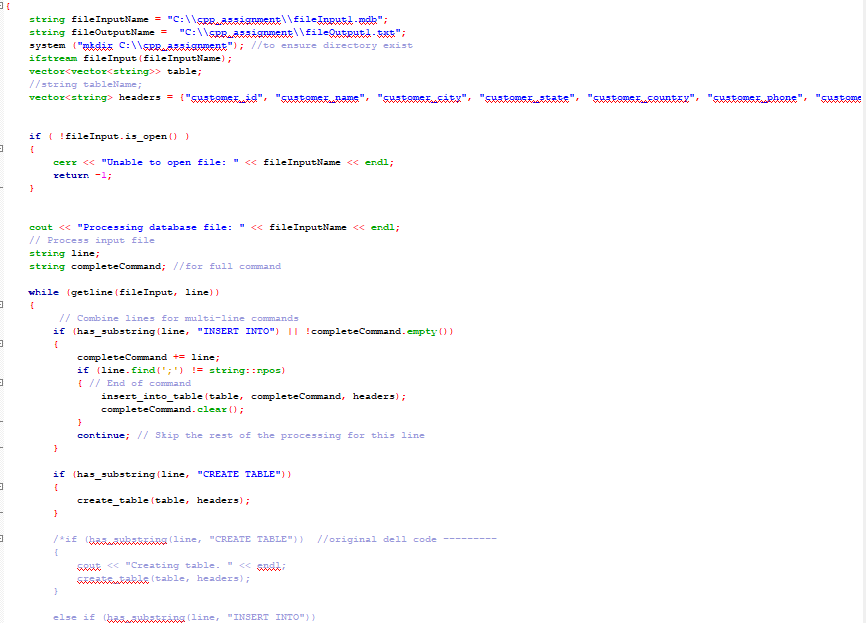




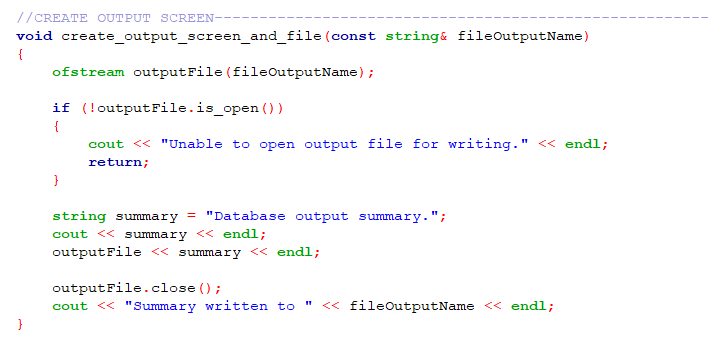
In the TT4L\_G01.cpp program, we link the fileInput1.mdb file to read database commands and process them. The results of these commands are written to fileOutput1.txt and displayed on the console.

* fileInput1.mdb , the file contains database commands, including table creation, data insertion, and data selection.
* fileOutput1.txt , The processed output is written into this file, including the database structure and data retrieval results

1. Reading from a file



1. OUTPUTTING TO A SCREEN



Pseudocode parts, explanations

|  |
| --- |
| START  1. DEFINE file paths:  - INPUT\_FILE = "C:\\cpp\_assignment\\fileInput1.mdb"  - OUTPUT\_FILE = "C:\\cpp\_assignment\\fileOutput1.txt"  2. OPEN input file (INPUT\_FILE) for reading:  IF file cannot be opened:  DISPLAY "Unable to open file"  EXIT program  3. READ each line from the input file:  WHILE there are lines to read:  STORE the current line in a variable (line)  DISPLAY the line (for debugging or processing confirmation)  4. CLOSE the input file.  5. WRITE the read data or processed output into the output file (OUTPUT\_FILE):  OPEN output file for writing.  WRITE necessary details based on input processing.  CLOSE the output file.  END |

This function reads data from a file, processes it, and saves the results to another file. It checks if the files can be opened, reads each line from the input, and writes the processed output to the specified location, ensuring everything is handled properly.

**Q03, Q09, Q11 [4] Update table rows, delete table rows, view table**

Screenshots (inputs, outputs), explanations

Pseudocode parts, explanations

|  |
| --- |
|  |

# **Q04, Q09, Q11 [2] Count and output number of rows in the table**

Screenshots (inputs, outputs), explanations

Pseudocode parts, explanations

|  |
| --- |
|  |

**Q05, Q11 [2] Must use vectors or arrays, functions or classes, to store file output contents**

Vectors used in creation of table to store input.

Code parts, explanations

|  |
| --- |
|  |

**Q06, Q11 [2] Inline comments, function or class comments, indentation, proper C++ naming with styling conventions**

Any violation is penalized by a reduction of 1 mark.

Screenshots (inputs, outputs), explanations

Code parts, explanations

|  |
| --- |
|  |

**Q07, Q09, Q11 [2] The program demonstrates error handlings**

[0: Below Expectation, 1: Within Expectation, 2: Exceed Expectation]

Screenshots (inputs, outputs), explanations

Pseudocode parts, explanations

|  |
| --- |
|  |

**Q08, Q11 [2] Structured diagrams**

Figures, explanations

**Q10, Q11 [3] Three sample input files, step by step screenshot outputs, output files, explanations**

Sample 1 for A1 (input file)

filename: fileInput1.mdb

|  |
| --- |
| CREATE fileOutput1.txt;  DATABASES;    CREATE TABLE customer(  customer\_id INT,  customer\_name TEXT,  customer\_city TEXT,  customer\_state TEXT,  customer\_country TEXT,  customer\_phone TEXT,  customer\_email TEXT  );  TABLES;    INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (1,'name1','city1','state1','country1','phone1','email1');  INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (2,'name2','city2','state2','country2','phone2','email2');  INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (3,'name3','city3','state3','country3','phone3','email3');  INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (4,'name4','city4','state4','country4','phone4','email4');    SELECT \* FROM customer; |

output file and screen output

filename: fileOutput1.txt

|  |
| --- |
| > CREATE fileOutput1.txt;  > DATABASES;  C:\mariadb\fileInput1.mdb  **>** CREATE TABLE customer(  customer\_id INT,  customer\_name TEXT,  customer\_city TEXT,  customer\_state TEXT,  customer\_country TEXT,  customer\_phone TEXT,  customer\_email TEXT  );  > TABLES;  customer  **>** INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (1,'name1','city1','state1','country1','phone1','email1');  > INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (2,'name2','city2','state2','country2','phone2','email2');  > INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (3,'name3','city3','state3','country3','phone3','email3');  > INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (4,'name4','city4','state4','country4','phone4','email4');  > SELECT \* FROM customer;  customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email  1,name1,city1,state1,country1,phone1,email1  2,name2,city2,state2,country2,phone2,email2  3,name3,city3,state3,country3,phone3,email3  4,name4,city4,state4,country4,phone4,email4 |

Sample 1 for A2 (input file)

filename: fileInput2.mdb

|  |
| --- |
| CREATE fileOutput2.txt;  DATABASES;    CREATE TABLE customer(  customer\_id INT,  customer\_name TEXT,  customer\_city TEXT,  customer\_state TEXT,  customer\_country TEXT,  customer\_phone TEXT,  customer\_email TEXT  );    INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (1,'name1','city1','state1','country1','phone1','email1');  INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (2,'name2','city2','state2','country2','phone2','email2');  INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (3,'name3','city3','state3','country3','phone3','email3');  INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (4,'name4','city4','state4','country4','phone4','email4');  SELECT \* FROM customer;    TABLES;    UPDATE customer SET customer\_email='email333' WHERE customer\_id=3;  SELECT \* FROM customer;    DELETE FROM customer WHERE customer\_id=4;  SELECT \* FROM customer;    SELECT COUNT(\*) FROM customer; |

output file and screen output

filename: fileOutput2.txt

|  |
| --- |
| > CREATE fileOutput2.txt;  > DATABASES;  C:\mariadb\fileInput2.mdb  **>** CREATE TABLE customer(  customer\_id INT,  customer\_name TEXT,  customer\_city TEXT,  customer\_state TEXT,  customer\_country TEXT,  customer\_phone TEXT,  customer\_email TEXT  );  **>** INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (1,'name1','city1','state1','country1','phone1','email1');  > INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (2,'name2','city2','state2','country2','phone2','email2');  > INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (3,'name3','city3','state3','country3','phone3','email3');  > INSERT INTO customer(customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email) VALUES (4,'name4','city4','state4','country4','phone4','email4');  > SELECT \* FROM customer;  customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email  1,name1,city1,state1,country1,phone1,email1  2,name2,city2,state2,country2,phone2,email2  3,name3,city3,state3,country3,phone3,email3  4,name4,city4,state4,country4,phone4,email4  > TABLES;  customer  > UPDATE customer SET customer\_email='email333' WHERE customer\_id=3;  > SELECT \* FROM customer;  customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email  1,name1,city1,state1,country1,phone1,email1  2,name2,city2,state2,country2,phone2,email2  3,name3,city3,state3,country3,phone3,email333  4,name4,city4,state4,country4,phone4,email4  > DELETE FROM customer WHERE customer\_id=4;  > SELECT \* FROM customer;  customer\_id,customer\_name,customer\_city,customer\_state,customer\_country,customer\_phone,customer\_email  1,name1,city1,state1,country1,phone1,email1  2,name2,city2,state2,country2,phone2,email2  3,name3,city3,state3,country3,phone3,email333  > SELECT COUNT(\*) FROM customer;  3 |

Sample 2

Input file covers all tasks, step by step screenshot outputs, output file, explanations

Your own sample?

Sample 3

Input file covers all tasks, step by step screenshot outputs, output file, explanations

Your own sample?