# Cover page

**Multimedia University**

**CCP6114 Programming Fundamentals 2430**

**Lecture section: ?TC2L / ?TC3L**

**Tutorial section: ?TT4L / ?TT5L / ?TT6L**

**Group number: ?04**

**Group leader student name: ?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Student**  **ID** | **Student Name by**  **alphabetical order** | **Task Descriptions** | **Percentage (%)** |
| 1 | 242UC244DD | Tiew Fu Siang | Create database, create table, insert into, select all |  |
| 2 | 242UC244PP | Nicholas Beh Zi Yang | Select count (where) pseudocode |  |
| 3 | 242UC24551 | Low Zheng Hao | Update (where) flowchart |  |
| 4 | 242UC244Q2 | Low Xuan Yu | Delete, input output screenshot |  |

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 | ? |
| Student name | ? |
| Task percentage | ? |
| Task descriptions | ? |
| Total score (30m) | ? |

Student 2

|  |  |
| --- | --- |
| Student ID | ? |
| Student name | ? |
| Task percentage | ? |
| Task descriptions | ? |
| Total score (30m) | ? |

Student 3

|  |  |
| --- | --- |
| Student ID | ? |
| Student name | ? |
| Task percentage | ? |
| Task descriptions | ? |
| Total score (30m) | ? |

Student 4

|  |  |
| --- | --- |
| Student ID | ? |
| Student name | ? |
| 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](#_Toc184561243)

[Mark sheet checklist (30%)](#_Toc184561244)

[Table of contents with page numbers and links](#_Toc184561245)

[Delete this information section](#_Toc184561246)

[Question Section](#_Toc184561247)

[Q01, Q09, Q11 [5] Database name, table name, table of two data types, insert table rows, view table in csv mode](#_Toc184561248)

[Q02, Q09, Q11 [3] Reading from a file, outputting to screen, writing to a file](#_Toc184561249)

[Q03, Q09, Q11 [4] Update table rows, delete table rows, view table](#_Toc184561250)

[Q04, Q09, Q11 [2] Count and output number of rows in the table](#_Toc184561251)

[Q05, Q11 [2] Must use vectors or arrays, functions or classes, to store file output contents](#_Toc184561252)

[Q06, Q11 [2] Inline comments, function or class comments, indentation, proper C++ naming with styling conventions](#_Toc184561253)

[Q07, Q09, Q11 [2] The program demonstrates error handlings](#_Toc184561254)

[Q08, Q11 [2] Structured diagrams](#_Toc184561255)

[Q10, Q11 [3] Three sample input files, step by step screenshot outputs, output files, explanations](#_Toc184561256)

# 

# Delete this information section

Insert the comment below at the beginning of your source code files:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Program: YOUR\_FILENAME.cpp

// Course: CCP6114 Programming Fundamentals

// Lecture Class: TC3L

// Tutorial Class: TT5L

// Trimester: 2430

// Member\_1: ID | NAME | EMAIL | PHONE

// Member\_2: ID | NAME | EMAIL | PHONE

// Member\_3: ID | NAME | EMAIL | PHONE

// Member\_4: ID | NAME | EMAIL | PHONE

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Task Distribution

// Member\_1:

// Member\_2:

// Member\_3:

// Member\_4:

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Question Section

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

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

Screenshots (inputs, outputs), explanations

?

Pseudocode parts, explanations

|  |
| --- |
|  |

## 

# 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

?

Pseudocode parts, explanations

|  |
| --- |
|  |

# 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

Screenshots (inputs, outputs), explanations

?

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?