Create a class CarUnit with the attribute:

std::string m_car_brand;
float m_car_price;
CarType m_car_type;(SEDAN,SUV,HATCHBACK)
int m_car_top_speed;
int m_car_rpm;
CarGear m_car_gear_system;(PRNDL,TIPTRONIC,CLASSIC)

Create a follwing function in a functionalities.cpp fie:

1) A function which Create at least 5 instance in a container which uses a unique std::syting ID attribute as key
with shared pointer to CarUnit instance being the value in the hashtable bucket.

2) A function which check wheather all instance in the data container have the same gear system or not (all
instance must have the same m_car_gear_system value, types should not taken as input from the user)

3) A function to return a container of instance whose m_car_type matches with the value passed as parameter to this function.

4) A function to find and return m_car_price of the instance whose m_car_top_speed is lowest.

5) A function to return the m_car_rpm value for all instance whose ID is passed as a parameter.(ID
should match with the key of the instance stored in the container)

6) A function to return the count of instance whose m_car_price is above a threshold value (given while
calliing the function as parameter)

Note:
1) Create special function as per requriment
2) All Corner Case for functionalities must be handled using std::nullopt and std::optional types.
3) All data processing operations must be performed using STL algorithms.
4) Demostrate functionalities by Createing a sutaible main.cpp file

create .h and .cpp file for each class and make a functionalities .h and functionalities.cpp and main.cpp

---

Create a class EvCar which having attribute.

Create a class Car having attribute:
std::string m_car_reg_num;
float m_price;
Reff m_car_engine; (using Reff=std::reference_wrapper<Engine>;)

Create a class Engine having attribute:

std::string m_car_id;
int m_battery_charging_time;
Platfrom m_platfrom; {PURE_EV or IC_BASED}
float m_chassos_length;
int m_seat_count;
int m_top_speed;
std::string m_engine_number;
int m_engine_torque;
int m_horsepower;
EngineType m_engine_type; (CRDI,MPFI,TURBOCHARGED)

Create following function in Functionalities.cpp file

1) A function Create at leat 5 instance in a container that stores data contiguously. instance must be created
on the heap and must be managed by smart pointer(std::shared_ptr)
Create the following functions in a singleton class Operation.

2) A function to find and return std::list of first N instance form the input container where N is taken as an input parameter.(Consider invalid N value corner case appropriately).
1) A function which create at least 5 instance in a container that allows index based access to instance and allows
dynamic resizing. instance must be craeted on the heap and must be managed by smart pointer(shared_ptr)

3) A function to find and return the m_top_speed of the EvCar instance form container whose m_car_id is passed as a parameter.
2) A function to return a container of Engine reference_wrapper for each instance of the input data. (Hint: Copy
m_car_engine attribute of all instance into a container and return it).

4) A function to find and return count of instance whose m_chassos_length is above 2.7f.
3) A function to return a boolean to indicate wheather all instance have m_engine_torque value above 120.

5) A function to return the container of all instance from the input data which satisfy a predicate. (passed as a fiunction pointer or lambda function).
4) A function to find the average m_horsepower value from instance whose m_price value is above a threashold value as a parameter.
[Hint: if threashold value is 45000, identify instance with m_price above 45000 and find average m_horsepower from these instance only.]

6) A function to find and return a boolean to indeicate wheather at leat one input data instance has m_platfrom value as a PURE_EV.

5) A function to find the m_engine_type value of the instance whose m_car_reg_num is passed as a parameter.

6) A function to find the return Engine reference_wrapper for the instance m_engine_torque is the highest.

Note:
a) Create special member function as per requriment.

b) All corner case for Functionalities must be handle using std::nullopt and std::optional
b) All Corner Cases for functionalities must be handled using std::optinal types. Customer exception are not required.

c) All data processing Operations must be performed using STL algorithms.

d) Demostrate functionalities by createing a suitable main.cpp file.

c) All data processing operation must be performed using STL algorithm.
e) Execute all functionalities in a seprate std::async thread

d) Demostrate Functionalities by createing a sutaible Main.cpp file

Create a .h and .cpp file for each class and Functionalities .h and .cpp and main.cpp file
Create .h and .cpp file for each class and .h and .cpp file for functionalities and main.cpp file also

---

create a class automobile with the following attributes:
Create a class EvCar which having attribute.

1) _id which is a string(assign a unique value for _id for each object)
2) _type which could be either REGULAR or TRANSPOSRT
3) _price which is float value.
4) _seat_count which is an interger
5) _engine_horsepower whcih is an interger
6) A function CalcuatlateGSt() which return 18% of the _price as answer.

std::string m_car_id;
    int m_battery_charging_time;
    Platfrom m_platfrom; {PURE_EV or IC_BASED}
    float m_chassos_length;
    int m_seat_count;
    int m_top_speed;

Create a EvCar which has following attributes
Create following function in Functionalities.cpp file

1) _battery_capacity whcih is a float
2) _price which is a float value

4) _charging_type which could be either DC,AC or BOTH.
5) A function CalcuatlateGSt() which return 10% of the _price as a answer
1) A function Create at leat 5 instance in a container that stores data contiguously. instance must be created
on the heap and must be managed by smart pointer(std::shared_ptr)

2) A function to find and return std::list of first N instance form the input container where N is taken as an input parameter.(Consider invalid N value corner case appropriately).

Create a singleton class Opeartion with the following member function.
3) A function to find and return the m_top_speed of the EvCar instance form container whose m_car_id is passed as a parameter.

1) A function to create at least 3 objects of type Automobile and at leat 2 object of type EvCar on the heap using
smart pointer in a vector type container. All instance created must be stored in the same container.
4) A function to find and return count of instance whose m_chassos_length is above 2.7f.

2) A function to find and display the average GST amount paid for all EvCar instance in the data container created.
. (Use the CalcuatlateGSt() function to fetch the amount).
5) A function to return the container of all instance from the input data which satisfy a predicate. (passed as a fiunction pointer or lambda function).

3) A function to find and display the count of Automobile instance whose _seat_count is above 4.
6) A function to find and return a boolean to indeicate wheather at leat one input data instance has m_platfrom value as a PURE_EV.

4) A function to find and display the average _price of all instance of Automobile in the data container.

5) A function to find if at least one instance of EvCar type has _charging_type value as DC or not and display true or false accordingly.
Note:
a) Create special member function as per requriment.

Note:
Execute all functionalities vy createing std::thread instance for each function in Opeartion class.
b) All corner case for Functionalities must be handle using std::nullopt and std::optional

Make a .h and .cpp file for each class and main.cpp file and .h file for eum class
c) All data processing operation must be performed using STL algorithm.

d) Demostrate Functionalities by createing a sutaible Main.cpp file

---

Create the following in Functionalities.cpp file
create a class automobile with the following attributes:

1) A function the accept a container of interger and return the sum of first 3 numbers in the container .
For Example, a container of values {1,2,3,4,5 } would return the value 6.
1) _id which is a string(assign a unique value for _id for each object)
2) _type which could be either REGULAR or TRANSPOSRT
3) _price which is float value.
4) _seat_count which is an interger
5) _engine_horsepower whcih is an interger
6) A function CalcuatlateGSt() which return 18% of the _price as answer.

2) A function that accept a container of interger and return the minium numbers amongst the last 3 numbers in the container.
For Example a containerof value{1,2,3,4,5} would return the value 3.

3) A functionto to accept a container of interger and return a container of all prime numbers form the given input value.
For Example a container of value {1,2,3,4,5} would return the value {2,3,5}
Create a EvCar which has following attributes

4) A function to accept a container of interger and a predicate lambda function. the function should return a container of interger
which pass the condition of the predicate (return true for predicate applied). For Example a container of value {1,2,3,4,5} along with a predicate
[](int num){return num%2==0;} would return {2,4}.
1) _battery_capacity whcih is a float
2) _price which is a float value
4) _charging_type which could be either DC,AC or BOTH.
5) A function CalcuatlateGSt() which return 10% of the _price as a answer

5) A function to accept a containerof string and return the string with the maxium amount of Characerts in it.
For Example a container os string {"C++" ,"Coding" , "Demo"} would return the value "Coding".

6) A function to accept a container of string and return a container of non-vowel Characerts from the input
string value. For Example , a container of string {"C++" ,"Coding" , "Demo"} would return
{'C' , '+' ,'+','C','d','n','g','D','m'}.
Create a singleton class Opeartion with the following member function.

1) A function to create at least 3 objects of type Automobile and at leat 2 object of type EvCar on the heap using
smart pointer in a vector type container. All instance created must be stored in the same container.

2) A function to find and display the average GST amount paid for all EvCar instance in the data container created.
. (Use the CalcuatlateGSt() function to fetch the amount).

3) A function to find and display the count of Automobile instance whose _seat_count is above 4.

4) A function to find and display the average _price of all instance of Automobile in the data container.

5) A function to find if at least one instance of EvCar type has _charging_type value as DC or not and display true or false accordingly.

Note:
Execute all functionalities vy createing std::thread instance for each function in Opeartion class.

Make a .h and .cpp file for each class and main.cpp file and .h file for eum class
---

Create a Vehicel with the following attribute:
Create the following in Functionalities.cpp file

_vehicel_milegae which is a float
_vehicel_fuel_capacity which is a float
_vehicel_cost which is a float
_vehicel_driver which is a refference wrapper to an instance of Driver created in stack memory
1) A function the accept a container of interger and return the sum of first 3 numbers in the container .
For Example, a container of values {1,2,3,4,5 } would return the value 6.

2) A function that accept a container of interger and return the minium numbers amongst the last 3 numbers in the container.
For Example a containerof value{1,2,3,4,5} would return the value 3.

creat a class Driver with the following attribute:
3) A functionto to accept a container of interger and return a container of all prime numbers form the given input value.
For Example a container of value {1,2,3,4,5} would return the value {2,3,5}

_driver_lience_type which of LMV,HMV or BOTH
_driver_experience_year which is an unsigned interger
_driver_rating which is a float and must be between 1 to 15 (This restriction must be applied to avoid objcet getting created with value beyond this range).
4) A function to accept a container of interger and a predicate lambda function. the function should return a container of interger
which pass the condition of the predicate (return true for predicate applied). For Example a container of value {1,2,3,4,5} along with a predicate
[](int num){return num%2==0;} would return {2,4}.

5) A function to accept a containerof string and return the string with the maxium amount of Characerts in it.
For Example a container os string {"C++" ,"Coding" , "Demo"} would return the value "Coding".

Create a functioalities.cpp section:
6) A function to accept a container of string and return a container of non-vowel Characerts from the input
string value. For Example , a container of string {"C++" ,"Coding" , "Demo"} would return
{'C' , '+' ,'+','C','d','n','g','D','m'}.

1) A function CreateObject to creat at leat 3 instance of Vehicel class on the heap using a shred_ptr

to Vehicel for each instance created.
2) A function CountInstance that can display the count of Vehicel instance whose _vehicel_milegae value is above threshold value provided as a parameter to the function.
3) A function to return a boolean to indicate whether all instance jhave an associated _driver_rating of above 4.0 or not
4) A function to find all instance whose _driver_lience_type is LMV and return a container of shred_ptr such Vehicel instance.
5) A function ComputeDriveSalary which display a value corresponding to each Vehicel instance based on the following
creta:

1) For a Vehicel instance whose associated Driver has _driver_rating above 4, value to be displayed should
be 4 times the _driver_experience_year value
2) For all other instance, value to be displayed should be 3 times _driver_experience_year
6) A function AddInstance that accepts all required data as function parameter and add a new instance to the
data container (Function should allow a user to add more instance to the data container by passing parameter to the function).

Note:

Note:

1. Demostrate each functioality vy creating sutiable client sie code.
2. Handle all possible cases.

Create a function FillData to generate an RXC dimension array of interger where R and C represent row and column
count respectively (taken as input form the user). Accept RXC inputs from the user and fill the array.
Create a Vehicel with the following attribute:

For eg: The input Sequence
2
2
10
20
30
40
_vehicel_milegae which is a float
_vehicel_fuel_capacity which is a float
_vehicel_cost which is a float
_vehicel_driver which is a refference wrapper to an instance of Driver created in stack memory

Should create this matrix of values

10 20
30 40

creat a class Driver with the following attribute:

_driver_lience_type which of LMV,HMV or BOTH
_driver_experience_year which is an unsigned interger
_driver_rating which is a float and must be between 1 to 15 (This restriction must be applied to avoid objcet getting created with value beyond this range).

Create an Adaptor function that accepts the following parameters
a) A value R represnting the count of rows in the 2D array.
b) A value C represnting the count of column in the 2D array.
c) A pointer to an array of interger(represnting the values generated in the FillData function).
d) A function wrapper to any function matching the following signature
1) 2 interger values (for R and C as descrived above) along with the pointer to an array of interger.
2) Return type of the function Should be void.

Create a functioalities.cpp section:

Invoke the Adaptor function by passing the following function as parameters to it
1) A function CreateObject to creat at leat 3 instance of Vehicel class on the heap using a shred_ptr
to Vehicel for each instance created.
2) A function CountInstance that can display the count of Vehicel instance whose _vehicel_milegae value is above threshold value provided as a parameter to the function.
3) A function to return a boolean to indicate whether all instance jhave an associated _driver_rating of above 4.0 or not
4) A function to find all instance whose _driver_lience_type is LMV and return a container of shred_ptr such Vehicel instance.
5) A function ComputeDriveSalary which display a value corresponding to each Vehicel instance based on the following
creta:

1) For a Vehicel instance whose associated Driver has _driver_rating above 4, value to be displayed should
be 4 times the _driver_experience_year value
2) For all other instance, value to be displayed should be 3 times _driver_experience_year
6) A function AddInstance that accepts all required data as function parameter and add a new instance to the
data container (Function should allow a user to add more instance to the data container by passing parameter to the function).

1) A function that takes the bove matrix as input, and display the sum of values in each row.

For E.g For the matrix above , console Should display
30
70

2) A function that display the higest value from the input matrix on the console.
Note:

For E.g For the matrix above, console Should display 40.

1. Demostrate each functioality vy creating sutiable client sie code.
2. Handle all possible cases.

3) Lambda Function that display the square of the number ayt the last position of the matrix.
For E.g For the matrix above, console Should display 1600

4) Lambda function that prints the maxium number in the each column of the matrix

For E.g For the Matrix above, console Should display

30
40

code in Mordern cpp

Note: Create the appropriate main function to demostrate all functioalities

Write code in morder cpp and Create a functioalities.h and functioalities.cpp and main.cpp files

---

creae a class Device having:
Create a function FillData to generate an RXC dimension array of interger where R and C represent row and column
count respectively (taken as input form the user). Accept RXC inputs from the user and fill the array.

_device_id which is a string
_device_type which is either INFOTAINMENT,ACCESSORY,SAFETY
_device_battery_level which must be an interger value between 1 to 100.
_device_driver which is a pointer to an instance of type DeviceDriver.
A fuction to find _batter_drain_factor which return a float value based on the floowing condition:
For eg: The input Sequence

2
2
10
20
30
40

Should create this matrix of values

1) if Device is of _device_type INFOTAINMENT or ACCESSORY , value sholud be 0.25.

10 20
30 40

2) Id Device is of _device_type SAFETY, value sholud be 0.5, if current _device_battery_level is above 50 else it sholud be 0.4.

Create an Adaptor function that accepts the following parameters
a) A value R represnting the count of rows in the 2D array.
b) A value C represnting the count of column in the 2D array.
c) A pointer to an array of interger(represnting the values generated in the FillData function).
d) A function wrapper to any function matching the following signature
1) 2 interger values (for R and C as descrived above) along with the pointer to an array of interger.
2) Return type of the function Should be void.

Create a class DeviceDriver with the following attributes.

1) _version_number which is a string value.
2) _release_quarter which could be Q1,Q2,Q3 or Q4.
3) _size_in_bytes which is a float value
Invoke the Adaptor function by passing the following function as parameters to it

Create a functionalites.cpp file:

1) A function to Create 5 instance of Device type on the heap using shared_ptr.
2) A function to find and return the _device_id of all instance in a container which return _batter_drain_factor value below 0.4
3) A function to check if all Device instance are of the same _device_type or not and return a boolean accordingly.
4) A function to display the lowest and heigst _size_in_bytes amongst instance whose _release_quarter is either Q1 or Q4 [Consider only Q1 and Q4 _release_quarter instance for computing the result].
5) A function to find and return the _version_number of all DeviceDriver for the instance whose _device_id matches with the id value provided as a parameter to the function.
6) A function to delete heap allocations if you have created instance using raw pointer
1) A function that takes the bove matrix as input, and display the sum of values in each row.

For E.g For the matrix above , console Should display
30
70

2) A function that display the higest value from the input matrix on the console.

For E.g For the matrix above, console Should display 40.

3) Lambda Function that display the square of the number ayt the last position of the matrix.
For E.g For the matrix above, console Should display 1600

4) Lambda function that prints the maxium number in the each column of the matrix

For E.g For the Matrix above, console Should display
30
40

Note: Create the appropriate main function to demostrate all functioalities

Write code in morder cpp and Create a functioalities.h and functioalities.cpp and main.cpp files

---

create a class Order
creae a class Device having:

std::string _id;
    float _value;
    OrderType _type;(PAID,COD,PROMOTION)
    float _discount;
_device_id which is a string
_device_type which is either INFOTAINMENT,ACCESSORY,SAFETY
_device_battery_level which must be an interger value between 1 to 100.
_device_driver which is a pointer to an instance of type DeviceDriver.
A fuction to find _batter_drain_factor which return a float value based on the floowing condition:

create a functionlitiea.cpp fie
1) if Device is of _device_type INFOTAINMENT or ACCESSORY , value sholud be 0.25.

1) A function to create 5 objcet on heap
2) Id Device is of _device_type SAFETY, value sholud be 0.5, if current _device_battery_level is above 50 else it sholud be 0.4.

2) a function to find and return the _type of the order whose _id is passed as an argument

3) A function and retun the _id of the order whose _discount amount is lowest. if there are mutltiple lowest value,consider the first lowest found form the start of the data contsiner.
Create a class DeviceDriver with the following attributes.

4) A function and retun the last N isnstance in the container where N is passed as a paramenter to function
1) _version_number which is a string value.
2) _release_quarter which could be Q1,Q2,Q3 or Q4.
3) _size_in_bytes which is a float value

Corner case requried
Create a functionalites.cpp file:

1) A function to Create 5 instance of Device type on the heap using shared_ptr.
2) A function to find and return the _device_id of all instance in a container which return _batter_drain_factor value below 0.4
3) A function to check if all Device instance are of the same _device_type or not and return a boolean accordingly.

4) A function to display the lowest and heigst _size_in_bytes amongst instance whose _release_quarter is either Q1 or Q4 [Consider only Q1 and Q4 _release_quarter instance for computing the result].
5) A function to find and return the _version_number of all DeviceDriver for the instance whose _device_id matches with the id value provided as a parameter to the function.
6) A function to delete heap allocations if you have created instance using raw pointer

---

create a class Order

std::string _id;
    float _value;
    OrderType _type;(PAID,COD,PROMOTION)
    float _discount;

create a functionlitiea.cpp fie

1) A function to create 5 objcet on heap

2) a function to find and return the _type of the order whose _id is passed as an argument

3) A function and retun the _id of the order whose _discount amount is lowest. if there are mutltiple lowest value,consider the first lowest found form the start of the data contsiner.

4) A function and retun the last N isnstance in the container where N is passed as a paramenter to function