EE402 Tutorial 2 – C and C++

Please solve the following short C++ problems. Your code should be concise and as efficient as possible, while still preserving best practice - i.e. avoid cutting-and-pasting any code. Note**: Each tutorial will be designed to have questions that become progressively more difficult, from fairly straightforward to very challenging.**

Q1. Write a function of the form void square(int &), where the int value that is passed by reference to the function is squared. Demonstrate that the function works correctly.

Q2. Blackjack: write a function of the form int blackjack(int, int) that when given 2 int values greater than 0, returns whichever value is nearest to 21 without going over. Return 0 if they both go over.

blackjack(19, 21) → 21
blackjack(21, 19) → 21
blackjack(19, 22) → 19

Q3. Write a class Vehicle that has the properties of a colour, make and model. Add a constructor for your class and add an abstract display() method.

Q4. Write a Car class that is a child of Vehicle (in Q3), which has the additional property of a number of seats. Add a suitable constructor and do everything necessary to allow you to create an object of the Car class.

Q5. Add an additional property to the Vehicle class (Q3, Q4) of a Vehicle Number. Use a static value (which starts at 10000) so that every vehicle has a unique vehicle number that it receives when the Vehicle is created. The Vehicle class is an abstract class so this will only happen through inheritance.

Q6. Add a destructor to the Car class and Vehicle class that displays the messages "A car has just been destroyed" and "A Vehicle has just been destroyed" respectively. Do something similar for the constructors of the two classes. Display what happens when you create and destroy an object of the Car class – do not use pointers. In one sentence, explain why this happens.

Q7. Perform the creation and destruction of the object using new and delete and see what happens. Add an inline accessor and mutator to the Vehicle class to get and set the colour of the Vehicle.

Warning – keep your Car class as you will need it in Question 10.

Q8. Write a function of the form: bool scoresIncreasing(int[], int) that when given an array of scores and the size of the array, returns true if each score is equal or greater than the one before. The array will be length 2 or more.

scoresIncreasing({1, 3, 4, 5, 6}, 5) → true
scoresIncreasing({1, 3, 2}, 3) → false
scoresIncreasing({1, 1, 4}, 3) → true

Q9. Write a function of the form bool hasOne(int) that when given a positive int, returns true if it contains a 1 digit. Note: use % to get the rightmost digit, and / to discard the rightmost digit.

hasOne(10) → true
hasOne(22) → false
hasOne(220) → false

Q10. Write a friend function of Car that has the form:

friend ostream &operator<<(ostream &stream, Car c);

Now write a suitable friend function of this form to display the details of the Car to the output stream. If this works correctly, in your main function you should be able to write:

cout << c; // where c is an object of the Car class.

Note: In your function, output to the stream value and not cout.