ADP STL map exercises

Some customers are allowed to pay later. This must now be part of the customer treatment. Thus, an individual credit maximum has to be stored for each customer and we will use a STL map object for handling this. Consult the book “beginning C++17” and the reference part of cplusplus.com for clues to solve the tasks below

Make a class Customer_credit_register. In this class declare as a private member a STL map of customer credit maximums. The key is the customer name and the credit maximum is just an integer.

Add a member function get_creditMax(string s), which returns the credit maximum for the customer (If the customer cannot be found, then -1 is returned) – you have to use the find member function to check that you have the customer in the map, before you return the value (why?). If a credit maximum cannot be shown, then the user has to call the member function register_creditMax_for_new_customer(string s, int c), which first checks whether the customer already is registered before inserting the new customer together with his/her maximum credit value into your map object (use the member function insert or just do an assignment).

Make a set_creditMax(string s, int c) member function for changing the credit value for the customer. Remember to check that the customer exists in the map before doing it.

Add a print member function.

Make a member function sum_of_creditMaxes(void), which returns the sum of all the creditMaxes.

Sometimes the shop is short of money, so make a member function clear_all_creditMaxes(void), which sets all the registered creditMaxes values to zero.

Inside the member function print make use of the function for_each from STL algorithms instead of the loop that you now have. For the third parameter of the for_each function you will need a lambda or a function object or just a function (try out all 3 options) for outputting a pair in the map. Notice that the parameter type is pair<string,int>. Declare the function object and the function above the class. (Remember to include algorithm: #include<algorithm>). In the reference at cplusplus.com you can find an example demonstrating the last 2 options.

Make a member function get_creditMax_maximum(void). Inside the function make use of the function max_element from STL algorithms. You need to supply a compare function or a functor or a lambda as the third parameter. Try out all 3 options. The return type must be bool. Notice that the parameter types are pair<string,int>. It returns true if the creditMax of the first pair parameter is less than the creditMax of the last pair. Write the function and the functor above the class.

Bonus round: Try to add more member function employing algorithms from <algorithm> wherever meaningful – just as you did with the algorithms for_each and max_element.