Assignment

Instructions

You are given a starter file filled with necessary stuff already. You just have to read each statement carefully and implement it. Submit .cpp file with filename set as your roll number and name. Example: MominaAtif-18P-0030

1 A Tale of Two Compositions

This is going to be a long story which doesn't mean a 'difficult' one. So, be patient.

1.1 Addresses

An organization typically keeps records of its employees' current and permanent addresses. Here's one way we can do this using classes:

```
class Employee {
        int current_address_house_no ;
        int current_address_street ;
        string current_address_city ;
        string current_address_country ;

        int permanent_address_house_no ;
        int permanent_address_street ;
        string permanent_address_city ;
        string permanent_address_country ;
};
```

 However, as you can see, this is a lot of repetition. So, we realize that we can define a new class that is called Address . This class will have a few public information pieces – house_no and street (which will be integers) and city and country (which will be strings).

1. Your first task is to define this class Address with all these pieces of information (and they should be publicly visible).

2. Also, you need to define a publicly visible function (get_full_address) within this class that returns the full address in the following format. (Values are given as an example – house_no is 4, street is 25 , city is Peshawar and country is Pakistan .)

H. No. 4, Street 25, Peshawar Pakistan
This function should return a string and not output the address line on the console!
(You may wish to use the included int_to_string() function to convert an int to string.)

 After this, you should be able to run Part - 1 of the main function in the starter file.

1.2 Employees

Once we have the Address class ready, you can define an Employee class.

1. This class should have the employee's name and two other variables of type Address
. One of these should be named current_address and the other permanent_address.

2. Also define a function (set_current_address) with the following declaration: void
set_current_address(int house_no, int street, string city, string country);In the body of
the function, you need to set the house_no, street, city and country of current_address
variable to the values provided in the arguments. (Hint: you can use the dot operator to
access or *assign* to the fields within the current_address)

3. Similar to the above, define another function as follows:
void set_permanent_address(int house_no, int street, string city, string
country); This would be exactly the same as above but will set the values within
permanent_address.

4. Finally, write another function (print_addresses()) that prints both addresses by calling
the function get_full_address() you defined above for the Address class. Do not try to
re-write the code you've already written.

After this, you should be able to run Part - 2 of the main function in the starter file.

2 Wheels and Cars

This part is very similar to the first one. So, if you got that, this should be straight forward. A
Car is composed of four Wheel thingies. If a car is moving, all its wheels are moving (if
nothing else has gone wrong).

2.1 Wheels for Cars

Let's first do wheels!

1. You need to define a Wheel class. At the moment, this class should have just a single
variable state of type string. Make sure this variable is hidden from those outside the

class!
2. Now, within this class, define a public function with the following declaration:
    void set_wheel_state(string s);
    This function should take set the state of the wheel to the value passed in as s.

3. Write another function in this class with the following declaration:
    string get_wheel_state()
    This should simply return the state of the wheel.
      After this, you should be able to run Part - 3 of the main function in the starter file.

2.2 Cars on Wheels

Once we have a Wheel class, we can use it within our Car class.

1. Define a new class Car . This should have an array of wheels – the size of the array should be 4 since we have only four wheels. Make sure this array is not visible from outside the class.

2. Next, write a function set_car_to_moving within this class. This function should set the states of all four wheels to "Turning" . That's it!

3. Write another function set_car_to_stopped same as above except the state of all four wheels should be set to "Stopped" .

4. Finally, write a function print_car_wheels_state in this class which outputs the state of all four wheels in the following format:

    Car state:
    - Wheel 0 is Stopped
    - Wheel 1 is Stopped
    - Wheel 2 is Stopped
    - Wheel 3 is Stopped

    Of course, if the car is stopped, all output lines should have the word Turning there. Make sure you use the
    get_wheel_state function of the wheel you defined earlier for this.

    After this, you should be able to run Part - 4 of the main function in the starter file.

And you're done!