Simulate an Instrument Cluster Data Display

Objective: Create a simplified data display system to simulate speed, fuel level, and engine temperature.

Requirements:

Implement a class-based design:

Create a VehicleData class to store parameters like speed, fuel level, and engine temperature.

Create a Display class to show these parameters on the console.

Simulate real-time updates:

Use a random number generator to update speed, fuel level, and temperature every second.

Display the updated data in a formatted manner (e.g., speed: 80 km/h, fuel: 50%, temperature: 90°C).

Add constraints:

Display warnings when parameters exceed thresholds (e.g., temperature > 100°C or fuel < 10%).

Deliverables:

A C++ program using multithreading (std::thread) to update and display data.

Output showcasing real-time updates and warnings.

Program:

#include <iostream>

#include <iomanip>

#include <random>

#include <thread>

#include <chrono>

#include <atomic>

using namespace std;

class VehicleData

{

public:

float speed;

float fuelLevel;

float engineTemp;

VehicleData() : speed(0), fuelLevel(100), engineTemp(90) {}

void updateData()

{

random\_device rd;

mt19937 gen(rd());

uniform\_real\_distribution<> speedDist(0, 180);

uniform\_real\_distribution<> fuelDist(0, 100);

uniform\_real\_distribution<> tempDist(70, 120);

speed = speedDist(gen);

fuelLevel = fuelDist(gen);

engineTemp = tempDist(gen);

}

};

class Display

{

public:

void showData(const VehicleData& data)

{

cout << "\rSpeed: " << fixed << setprecision(1) << data.speed

<< " km/h | Fuel: " << fixed << setprecision(1) << data.fuelLevel

<< "% | Temperature: " << fixed << setprecision(1) << data.engineTemp

<< "°C";

if (data.engineTemp > 100)

{

cout << " | WARNING: High Temperature!";

}

if (data.fuelLevel < 10)

{

cout << " | WARNING: Low Fuel!";

}

cout.flush();

}

};

void updateVehicleData(VehicleData& data)

{

while (true)

{

data.updateData();

this\_thread::sleep\_for(chrono::seconds(1));

}

}

void displayVehicleData(const VehicleData& data, Display& display)

{

while (true)

{

display.showData(data);

this\_thread::sleep\_for(chrono::seconds(1));

}

}

int main()

{

VehicleData vehicleData;

Display display;

thread updateThread(updateVehicleData, ref(vehicleData));

thread displayThread(displayVehicleData, cref(vehicleData), ref(display));

updateThread.join();

displayThread.join();

return 0;

}

Output:

Speed: 147.6 km/h | Fuel: 65.5% | Temperature: 110.8°C | WARNING: High Temperature! WARNING: Low Fuel!