**IMPLEMENTATION**

**C++ Code:**

#include <iostream>

#include <vector>

#include <string>

#include <ctime>

#include <iomanip>

using namespace std;

enum Status {

OPERATIONAL,

FAULTY,

UNDER\_MAINTENANCE

};

class Equipment {

private:

string name;

Status status;

float temperature;

float pressure;

public:

Equipment(string name) : name(name), status(OPERATIONAL), temperature(25.0f), pressure(1.0f) {}

void simulatePerformance() {

temperature += (rand() % 5 - 2);

pressure += (rand() % 3 - 1);

if (temperature > 50.0f) {

status = FAULTY;

} else if (pressure > 3.0f) {

status = FAULTY;

} else if (temperature < 20.0f) {

status = UNDER\_MAINTENANCE;

} else {

status = OPERATIONAL;

}

}

void displayStatus() {

cout << "Equipment: " << name << endl;

cout << "Status: " << statusToString() << endl;

cout << "Temperature: " << fixed << setprecision(2) << temperature << " °C" << endl;

cout << "Pressure: " << fixed << setprecision(2) << pressure << " bar" << endl;

cout << "-------------------------------------------" << endl;

}

string statusToString() {

switch (status) {

case OPERATIONAL: return "Operational";

case FAULTY: return "Faulty";

case UNDER\_MAINTENANCE: return "Under Maintenance";

default: return "Unknown";

}

}

string getName() {

return name;

}

Status getStatus() {

return status;

}

};

class EquipmentMonitor {

private:

vector<Equipment> equipmentList;

public:

void addEquipment(string name) {

equipmentList.push\_back(Equipment(name));

}

void monitor() {

cout << "Starting equipment monitoring system..." << endl;

while (true) {

cout << "\nMonitoring Equipment Status..." << endl;

for (auto & equipment : equipmentList) {

equipment.simulatePerformance();

equipment.displayStatus();

}

cout << "-------------------------------------------" << endl;

cout << "Press Enter to continue or type 'q' to quit: ";

string userInput;

getline(cin, userInput);

if (userInput == "q" || userInput == "Q") {

break;

}

}

}

};

int main() {

EquipmentMonitor monitor;

monitor.addEquipment("Pump A");

monitor.addEquipment("Compressor B");

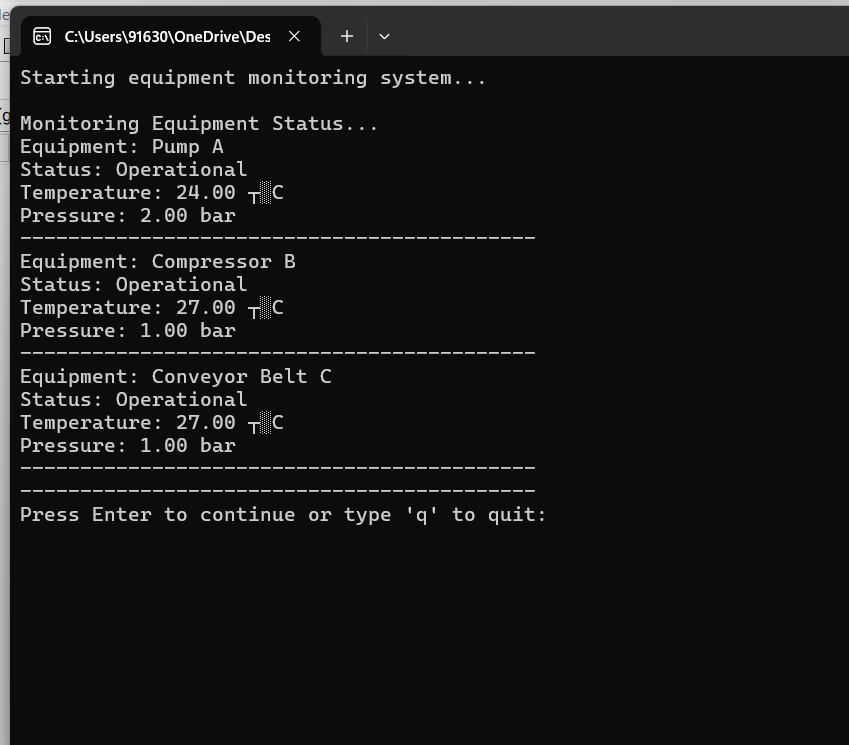
monitor.addEquipment("Conveyor Belt C");

monitor.monitor();

return 0;

}

**OUTPUT**

****