* Sprint-1

#include <iostream>

#include <string>

class HomeOwner {

private:

std::string name;

public:

HomeOwner(const std::string& name) : name(name) {}

std::string getName() const { return name; }

};

class EnvironmentSensor {

private:

std::string season;

int temperature;

public:

EnvironmentSensor(const std::string& season, int temperature) : season(season), temperature(temperature) {}

std::string getSeason() const { return season; }

int getTemperature() const { return temperature; }

};

class WaterHeater {

private:

int temperature;

public:

WaterHeater() : temperature(0) {}

void setTemperature(int temp) { temperature = temp; }

int getTemperature() const { return temperature; }

};

class SmartGeyser {

private:

HomeOwner owner;

EnvironmentSensor sensor;

WaterHeater heater;

public:

SmartGeyser(const HomeOwner& owner, const EnvironmentSensor& sensor) : owner(owner), sensor(sensor) {}

void activate() {

int currentTemp = sensor.getTemperature();

int requiredTemp = currentTemp + 15;

heater.setTemperature(requiredTemp);

std::cout << "Geyser activated. Heating temperature set to: " << requiredTemp << " degrees Celsius.\n";

}

};

int main() {

HomeOwner owner("RAM");

EnvironmentSensor sensor("Winter", 5);

SmartGeyser geyser(owner, sensor);

geyser.activate();

return 0;

}

* Sprint-2

#include <iostream>

#include <string>

class HomeOwner {

private:

std::string name;

public:

HomeOwner(const std::string& name) : name(name) {}

std::string getName() const { return name; }

};

class EnvironmentSensor {

private:

std::string season;

int temperature;

public:

EnvironmentSensor(const std::string& season, int temperature) : season(season), temperature(temperature) {}

std::string getSeason() const { return season; }

int getTemperature() const { return temperature; }

};

class WaterHeater {

private:

int temperature;

public:

WaterHeater() : temperature(0) {}

void setTemperature(int temp) { temperature = temp; }

int getTemperature() const { return temperature; }

};

class SmartGeyser {

private:

HomeOwner owner;

EnvironmentSensor sensor;

WaterHeater heater;

public:

SmartGeyser(const HomeOwner& owner, const EnvironmentSensor& sensor) : owner(owner), sensor(sensor) {}

void activate() {

std::cout << "Geyser activated. Enjoy your warm shower!\n";

}

};

int main() {

HomeOwner owner("RAM");

EnvironmentSensor sensor("Winter", 10);

SmartGeyser geyser(owner, sensor);

geyser.activate();

return 0;

}