

```

#include <cstdlib>
#include <initializer_list>
#include <memory>
#include <string>
#include <iostream>

using std::string, std::to_string, std::shared_ptr, std::make_shared;

// Eatable
class Eatable {
public:
    virtual string description()=0;
    virtual float cost()=0;
// Usine
    enum Food { lasagna, pasta, mozza, peperroni };
    static shared_ptr<Eatable> makeEatable(std::initializer_list<Food> foods);
};

// Meal
class Lasagna: public Eatable {
public:
    virtual string description() override { return "Lasagna " + to_string(this->cost()) + "$"; }
    float cost() override { return 5; }
};

class Pasta: public Eatable {
public:
    virtual string description() override { return "Pasta " + to_string(this->cost()) + "$"; }
    float cost() override { return 4; }
};

// Topping
class Topping: public Eatable {
protected:
    const shared_ptr<Eatable> base_meal;
public:
    Topping(shared_ptr<Eatable> base_meal): base_meal{base_meal} {}
    virtual float cost() override { return this->base_meal->cost() + 1; }
};

class Mozza : public Topping {
public:
    using Topping::Topping;
    string description() override { return this->base_meal->description() + " Mozza " + to_string(this->cost()) + "$"; }
};

class Peperroni : public Topping {
public:
    using Topping::Topping;
    string description() override { return this->base_meal->description() + " Peperroni " + to_string(this->cost()) + "$"; }
};

shared_ptr<Eatable> Eatable::makeEatable(std::initializer_list<Food> foods) {
    shared_ptr<Eatable> meal;
    auto item = foods.begin();
    switch(*item) {
        case lasagna: meal = make_shared<Lasagna>();break;
        case pasta: meal = make_shared<Pasta>();break;
        default: throw "No Meal";
    }

    for (++item; item != foods.end(); ++item) {
        switch(*item) {
            case mozza: meal = make_shared<Mozza>(meal);break;
            case peperroni: meal = make_shared<Peperroni>(meal);break;
            default: throw "Not a topping";
        }
    }
    return meal;
}

int main() {
    std::cout << Eatable::makeEatable({Eatable::lasagna, Eatable::peperroni})->description() << std::endl;
    std::cout << make_shared<Mozza>(make_shared<Pasta>())->description() << std::endl;
    return 0;
}

```