```
class  Baker {
    static const int INITIAL_SIZE = 20;
    friend ostream& operator<<(ostream& os, const Baker& baker);

private:
    String name;
    Treat** treats;
    size_t size;
    size_t capacity;
public:
    Baker(const string& name): name(name), treats(nullptr), size(0), capacity(0)
                                                                            {}
    ~Baker()
    Baker(const Baker& rhs);
    void bakes(const string& treatName){
        if(treats == nullptr){
            treats = new Treat*[INITIAL_SIZE];
            capacity = INITIAL_SIZE;
        } else if (size == capacity){
            Treat** newCollection = new Treat*[capacity*2]
            for (size_t i=0; i<size; ++i){
why?            newCollection[i] = new Treat(*(treats[i]));
                delete treats[i];
            }
            delete [] treats;
            capacity *= 2;
            treats = newCollection;
        }
        treats[size] = new Treat(treatName);
        ++size;
    }
```