

Item 5: Maximally Reusable Generic Containers - Part 2

Le Van Huynh

October 22, 2023

Contents

1 Question

1

1 Question

What is the following solution doing, and why? Explain each constructor and operator. Does this design or code have any flaws?

```
template <typename T, size_t size>
class fixed_vector {
public:
    typedef T* iterator;
    typedef const T* const_iterator;

    fixed_vector() { }

    template <typename O, size_t osize>
    fixed_vector(const fixed_vector<O, osize>& other)
    {
        copy(other.begin(),
            other.begin() + min(size, osize),
            begin());
    }

    template <typename O, size_t osize>
    fixed_vector<T, size>&
```

```

operator=(const fixed_vector<T, osize>& other)
{
    copy(other.begin(),
        other.begin() + min(size, osize),
        begin());
    return *this;
}

iterator begin() { return v_; }

iterator end() { return v_ + size; }

const_iterator begin() const { return v_; }

const_iterator end() const { return v_ + size; }

private:
    T v_[size];
};

```