1 Slide 4.5

```
struct positive_integer
{
  void set_value(int _v)
  {
    if(_v >= 0)
    {
      value = _v;
    }
  }
  int get_value() const
  {
    return value;
  }
  int value;
};

  // principle of least privilege violation
  positive_integer pi;
  pi.value = -2;

  // Benefit of const
  void edit_value(const positive_integer& _value)
  {
    _value .set_value(12);
  }
```

2 Slide 4.12

```
template<typename _T_>
class nodes_list
{
public:
  nodes_list();
  position* first();
  position* last();
  position* insertBefore(position* p, _T_ v);
  void remove(position* p);
};
```

Position structure

```
struct position
{
  _T_ value;
  position* next;
  position* previous;
};
```

implement first / last / constructors / insertBefore / remove

3 Slide 4.14

```
std::vector<int> fun();
std::vector<int> v3(3);
std::cout << v3.size() << std::endl;

std::vector<int> v4(std::move(v3));
std::cout << v4.size() << "␣" << v3.size() << std::endl;
```

4 Slide 4.20

```
template<typename _C_, typename _T_>
class stack_adater
{
public:
  stack_adater();
  std::size_t size() const;
  bool empty() const;
  _T_ top() const;
  _T_ push() const;
  _T_ pop() const;
};
```