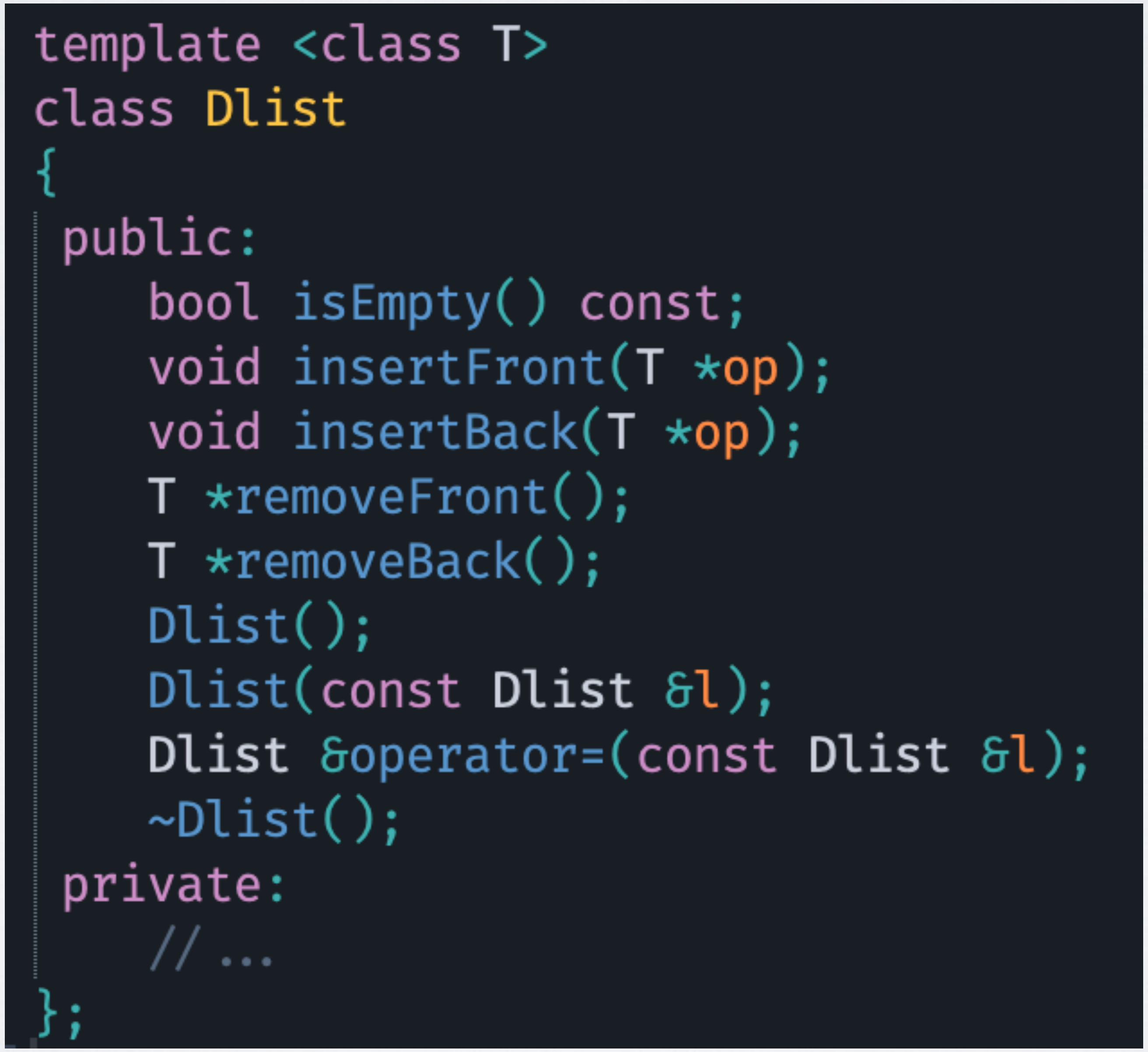
INTRO. TO PROGRAMMING FINAL

Zekai Fan
August 4, 2019

Template

- A special class where the data type is a parameter.



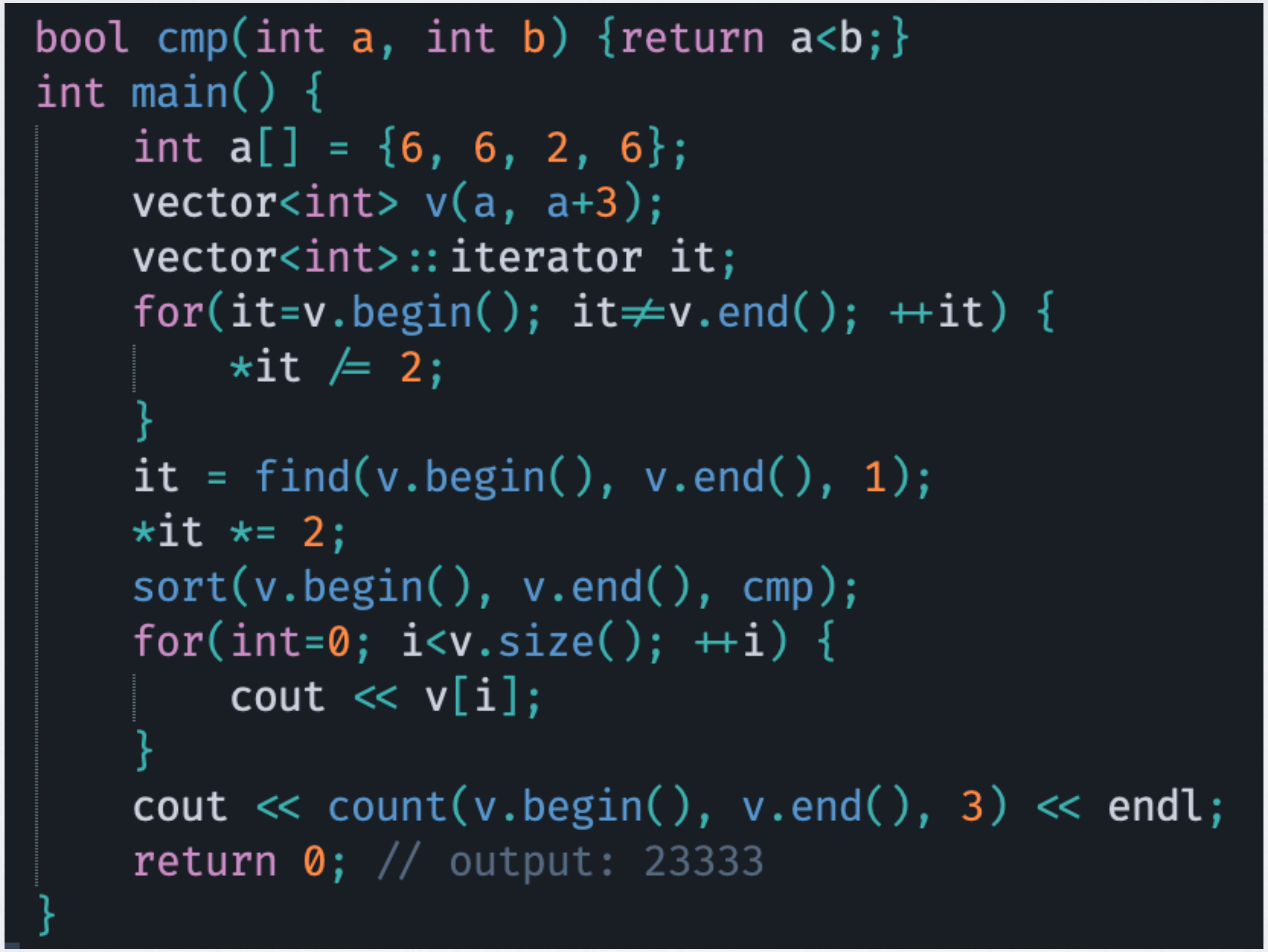
```
template <class T>
class Dlist
{
 public:
    bool isEmpty() const;
    void insertFront(T *op);
    void insertBack(T *op);
    T *removeFront();
    T *removeBack();
    Dlist();
    Dlist(const Dlist &l);
    Dlist &operator=(const Dlist &l);
    ~Dlist();
 private:
    // ...
};
```

Standard Template Library - Sequence Containers

- Vector, Stack & Queue (Homework 8 ex. 1)
  - Vector
    - size(), push_back(x), pop_back(), front(), back()
  - Stack (First in Last out) //order reversed
    - size(), push(x), pop(), top()
  - Queue (First in first out) //order unchanged
    - size(), push(x), pop(), front(), back()

Standard Template Library - Sequence Containers

- Iterator: pointers to the elements of the container



```
bool cmp(int a, int b) {return a<b;}
int main() {
    int a[] = {6, 6, 2, 6};
    vector<int> v(a, a+3);
    vector<int>::iterator it;
    for(it=v.begin(); it!=v.end(); ++it) {
        *it /= 2;
    }
    it = find(v.begin(), v.end(), 1);
    *it *= 2;
    sort(v.begin(), v.end(), cmp);
    for(int=0; i<v.size(); ++i) {
        cout << v[i];
    }
    cout << count(v.begin(), v.end(), 3) << endl;
    return 0; // output: 23333
}
```

GOOD LUCK!