

Streams and operators

Jack Garner

October 1, 2019

Most built in operators can be overloaded

```
1  vector<int> operator+(vector<int> a, vector<int> b)  {  
2      for (auto x : b) {  
3          a.push_back(x);  
4      }  
5      return a;  
6  }
```

Most built in operators can be overloaded

```
1  vector<int> operator+(vector<int> a, vector<int> b)  {  
2      for (auto x : b) {  
3          a.push_back(x);  
4      }  
5      return a;  
6  }
```

<https://en.cppreference.com/w/cpp/language/operators>

Try to print a vector

```
1  int main() {  
2      vector<int> v1{0, 1, 2};  
3      vector<int> v2{3, 4, 5};  
4      cout << (v1 + v2) << endl; // :(  
5      return 0;  
6  }
```

Try to print a vector

```
1  int main() {  
2      vector<int> v1{0, 1, 2};  
3      vector<int> v2{3, 4, 5};  
4      cout << (v1 + v2) << endl; // :(  
5      return 0;  
6  }
```

Can we overload <<

Try to print a vector

```
1  ostream& operator<<(ostream& os, const vector<int>& b) {
2      os << "[";
3      for (auto x : b) {
4          os << x << ",";
5      }
6      os << "]" ;
7      return os;
8  }
9
10 int main() {
11     vector<int> v1{0, 1, 2};
12     vector<int> v2{3, 4, 5};
13     cout << (v1 + v2) << endl; // :)
14     return 0;
15 }
```

Plenty of things are streams in C++

- Reading/Writing to the console
- Files
- Networks
- Strings
- Anything you want (with some work)

Streams

Plenty of things are streams in C++

- Reading/Writing to the console
- Files
- Networks
- Strings
- Anything you want (with some work)

```
1 ifstream readOnly("readMe.txt");
2 ofstream writeOnly("logger.txt");
3 fstream readWrite("text.txt");
4 writeOnly << "t" << endl;
5 int x;
6 readOnly >> x;
```


Try It!

Complex numbers contain both a real and complex component. Try creating a class to represent them. Next, check out all of the operators you can define and see how many of them you can write implementations for.