

```
1: // $Id: for_loop.cpp,v 1.24 2014-06-27 13:08:31-07 - - $
2:
3: //
4: // Simple class range which allows a range of integral values
5: // to be used in a foreach loop in the same way as iterators
6: // may be used.
7: //
8:
9: #include <iostream>
10: using namespace std;
11:
12: template <typename Discrete> class range;
13: template <typename Discrete>
14: ostream& operator<< (ostream& out, const range<Discrete>& range_) {
15:     return out << "range(" << range_.begin_ << "," << range_.end_ << ")";
16: }
17:
18: template <typename Discrete>
19: class range {
20:     private:
21:         Discrete begin_ {};
22:         Discrete end_ {};
23:     public:
24:         range() {}
25:         range (Discrete begin, Discrete end): begin_(begin), end_(end) {}
26:         class iterator;
27:         iterator begin() { return {begin_}; }
28:         iterator end() { return {end_}; }
29:         friend ostream& operator<< <> (ostream&, const range<Discrete>&);
30: };
31:
32: template <typename Discrete>
33: class range<Discrete>::iterator {
34:     private:
35:         friend class range<Discrete>;
36:         Discrete value {};
37:         iterator (Discrete value): value(value) {}
38:     public:
39:         iterator() {}
40:         Discrete operator*() { return value; }
41:         iterator operator++() { ++value; return *this; }
42:         Discrete operator!= (range::iterator other) {
43:             return value != other.value;
44:         }
45: };
46:
```

```
47:
48: int main (int, char**) {
49:     range<int> integers (1, 11);
50:     cout << "integers range is " << integers << endl;
51:     for (int val: integers) {
52:         cout << (val != *integers.begin() ? " " : "") << val;
53:     }
54:     cout << endl;
55:     for (range<int>::iterator itor = integers.begin();
56:         itor != integers.end(); ++itor) {
57:         cout << (itor != integers.begin() ? " " : "") << *itor;
58:     }
59:     cout << endl;
60:     range<char> chars (' ', '~' + 1);
61:     cout << "chars range is " << chars << endl;
62:     for (char chr: chars) {
63:         cout << chr;
64:         if ((chr & 0x1F) == 0x1F) cout << endl;
65:     }
66:     cout << endl;
67:     return 0;
68: }
69:
70: //TEST// ./for_loop >for_loop.out 2>&1
71: //TEST// mkpspdf for_loop.ps for_loop.cpp* for_loop.out
72:
```

[illegible]

```
1: integers range is range(1,11)
2: 1 2 3 4 5 6 7 8 9 10
3: 1 2 3 4 5 6 7 8 9 10
4: chars range is range( ,\177)
5: !"#$%&'()*+,-./0123456789:;<=>?
6: @ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_
7: `abcdefghijklmnopqrstuvwxyz{|}~
```