



```
1
2 use std::env;
3 use std::fmt;
4
5 // pub trait Display {
6 //     fn fmt(&self, f: &mut Formatter<'_>) -> Result<(), Error>;
7 // }
8
9 #[derive(Copy, Clone, Debug)]
10 struct Point { x: f64, y: f64, }
11
12 #[derive(Copy, Clone, Debug)]
13 struct LatLong { ns: f64, ew: f64, }
14
15 impl fmt::Display for LatLong {
16     fn fmt(&self, f: &mut fmt::Formatter<'_>) -> fmt::Result {
17         let mut ewc = 'E'; if self.ew < 0.0 { ewc = 'W'; }
18         let mut nsc = 'N'; if self.ns < 0.0 { nsc = 'S'; }
19         write!(f, "({}{}, {}{})", self.ew, ewc, self.ns, nsc)
20     }
21 }
22
23 fn main() {
24
25     println!("Traits examples");
26     let mut p1 = Point{x:0.0,y:0.0};
27     let mut p2 = Point{x:1.0,y:1.0};
28     p2 = p1;
29     println!("p1 = {:?}",p1);
30     println!("p2 = {:?}",p2);
31
32     let pos = LatLong{ew:102.0,ns:14.3};
33     println!("Pos {} ", pos );
34
35     println!("Traits examples");
36     let mut p1 = Point{x:0.0,y:0.0};
37     let mut p2 = Point{x:1.0,y:1.0};
38     p2 = p1;
39     println!("p1 = {:?}",p1);
40     println!("p2 = {:?}",p2);
41 }
```

