

Lecture 11

Thursday Oct. 12

```

class Directory {
    void addFile (String fn) {
        File nf = new File (fn);
        :
    }
}

```

```

} void addFile (File f) {
}
File fl = new File ("fz.txt");
d1.addFile (f1);

```

```

Directory d1 = new Dir ("D");
d1.addFile ("fl.txt");

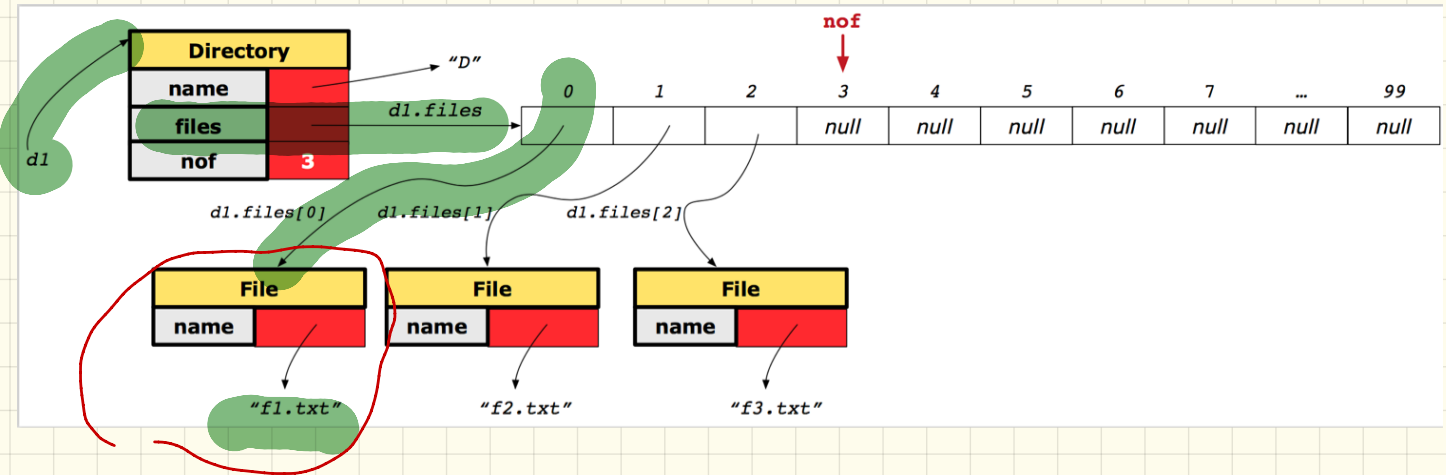
```

```

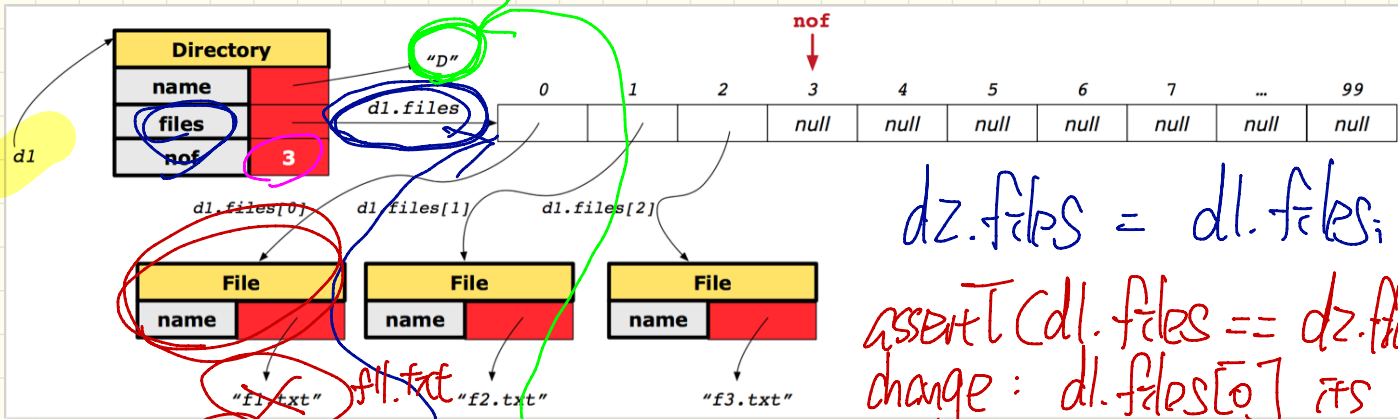
Directory dz =
    new Directory (...);
Supplier
dz.addFile (f);

```

violation of composition (sharing) they have no reference to the client new File object created



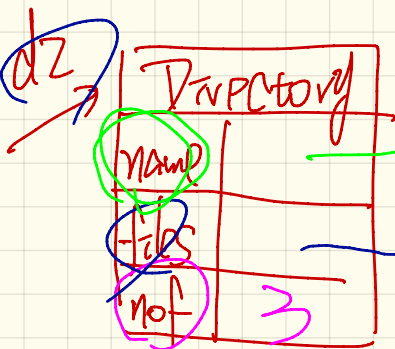
this file object
 is only referenced
 from d1.
 (no sharing via
 aliasing).



`dz.files = d1.files;`

`assert(d1.files == dz.files)`
 change: `d1.files[0]` its name to "f1.txt"

`Directory dz = new Directory(d1);`



class Directory {
 Directory(Directory other) {
 printout dz
 this.nof = other.nof;
 this.files = other.files;
 this.name = other.name;
 }
 dz
 other
 a copy of d1.
 call by ref.

```
class CEmployee implements Comparable<> {
```

```
    int compareTo (CEmployee other) {
```

```
        return this.id - other.id;
```

alan

other

```
} After Arrays.sort : tom, alan, mark → mark.compareTo(alan) |
```

CEmployee objects :
alan
mark
tom

ids
②
③
1

JAVA

alan.compareTo(mark) - |
tom.compareTo(alan) - |

math

alan < mark
tom < alan

tom < mark

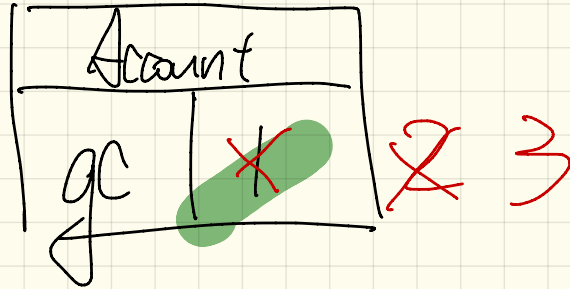
alan 5000

mark 3000

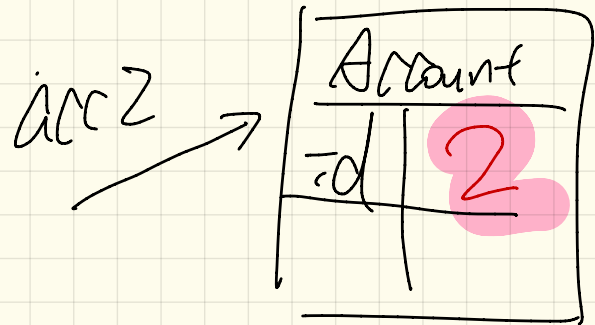
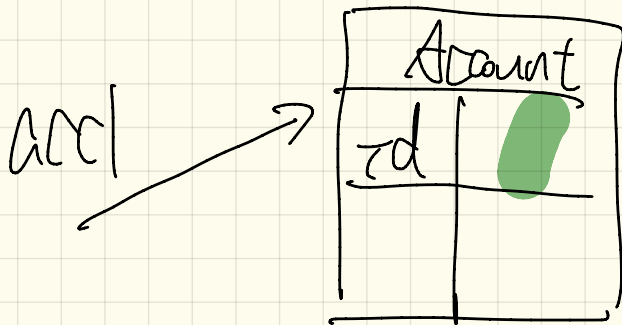
Double. compare (alan.salary, mark.salary)

+ returning this as the
result for compare in
Employee2 is not right

static int globalCounter



acc1 = new Account("Jim");



```
class Account {  
    String branchName;
```

```
    static int c;  
    static void m() {
```

```
        branchName
```

```
    }
```

```
Account m()
```

```
Account acc =  
    new Account("B");  
acc.branchName;
```