Custom Data Structures

Iterable

```java
// old way
for (int i = 0; i < stars.size(); i++) {
    Star s = stars.get(i);
    System.out.println(s);
    telescope.scheduleTime(s);
    researchFunds.allocate(s);
}

// new way
for (Star s: stars) {
    System.out.println(s);
    telescope.scheduleTime(s);
    researchFunds.allocate(s);
}
```

Enhanced For Loop

- Implement interface `java.lang.Iterable`
- Implement Iterator

```java
for (Student s: myRoster) {
    Systemout.println(s)
}
// compiles to below code (implement this code for custom iterator)
Iterator<Student> it = myRoster.iterator();
while (it.hasNext()) {
    Student s = it.next();
    System.out.println(s);
}

// a simple way for implementation

public Iterator<Student> iterator {
    return collectToList().iterator();
}
```

Loop Safety

- Do not mutate list directly from the loop
- Collect items to be removed and remove them outside the loop

TreeSet

- Iterator uses comparator

ArrayList, LinkedList

- FIFO

HashSet

- Arbitrary order

Example

```
@Override
public boolean equals(Object o) {...}

@Override
public int hashCode() {...}

@Override
public int compareTo(Student s) {...}
```

Generic

- `T`