Agenda

- Comparable
- Comparator
- Collection FrameWork
- ~~Traversal~~
- ~~FailSafe and FailFast Iterator~~
- ~~List~~

Generic Interfaces

- Interface is standard/specification.
- comparable is a predefined interface in java

```
// Comparable is pre-defined interface which was non-generic till Java 1.4

interface Comparable {
int compareTo(Object obj);
}

class Person implements Comparable {
// ...
public int compareTo(Object obj) {
Person other = (Person)obj; // down-casting
// compare "this" with "other" and return difference
}
}

class Program {
  public static void main(String[] args) {
    Person p1 = new Person("James Bond", 50);
    Person p2 = new Person("Ironman", 45);
    int diff = p1.compareTo(p2);
    if(diff == 0)
    System.out.println("Both are same");
    else if(diff > 0)
    System.out.println("p1 is greater than p2");
    else //if(diff < 0)
    System.out.println("p1 is less than p2");
    diff = p2.compareTo("Superman"); // will fail at runtime with
ClassCastException (in down-casting)
}
}
```

- Generic interface has type-safe methods (arguments and/or return-type).

```
// Comparable is pre-defined interface -- generic since Java 5.0
interface Comparable<T> {
```

```
int compareTo(T obj);
}

class Person implements Comparable<Person> {
// ...
public int compareTo(Person other) {
// compare "this" with "other" and return difference
}
}

class Program {
  public static void main(String[] args) {
    Person p1 = new Person("James Bond", 50);
    Person p2 = new Person("Ironman", 45);
    int diff = p1.compareTo(p2);
    if(diff == 0)
    System.out.println("Both are same");
    else if(diff > 0)
    System.out.println("p1 is greater than p2");
    else //if(diff < 0)
    System.out.println("p1 is less than p2");
    diff = p2.compareTo("Superman"); // compiler error
}
}
```

Comparable<>

- Standard for comparing the current object to the other object.
- Has single abstract method int compareTo(T other);
- In java.lang package.
- Used by various methods like Arrays.sort(Object[]), ...
- It does the comparision for the natural ordering

Comparator<>

- Standard for comparing two (other) objects.
- Has single abstract method int compare(T obj1, T obj2);
- In java.util package.
- Used by various methods like Arrays.sort(T[], comparator), ...

Collection Framework

- Collection framework is Library of reusable data structure classes that is used to develop application.
- Main purpose of collection framework is to manage data/objects in RAM efficiently.
- Collection framework was introduced in Java 1.2 and type-safe implementation is provided in 5.0 (using generics).
- Collection is available in java.util package.
- Java collection framework provides

1. Interfaces -- defines standard methods for the collections.

2. Implementations -- classes that implements various data stuctures.
3. Algorithms -- helper methods like searching, sorting, ...

Collection Hierarchy

- Interfaces: Iterable, Collection, List, Queue, Set, Map, Deque, SortedSet, SortedMap, ...
- Implementations: ArrayList, LinkedList, HashSet, HashMap, ...
- Algorithms: sort(), reverse(), max(), min(), ... -> in Collections class static methods

Collection interface

- Root interface in collection framework interface hierarchy.
- Most of collection classes are inherited from this interface (indirectly).
- Provides most basic/general functionality for any collection
- Abstract methods
    - boolean add(E e)
    - int size()
    - boolean isEmpty()
    - void clear()
    - boolean contains(Object o)
    - boolean remove(Object o)
    - boolean addAll(Collection<? extends E> c)
    - boolean containsAll(Collection<?> c)
    - boolean removeAll(Collection<?> c)
    - boolean retainAll(Collection<?> c)
    - Object[] toArray()
    - Iterator iterator() -- inherited from Iterable
- Default methods
    - default Stream stream()
    - default Stream parallelStream()
    - default boolean removeIf(Predicate<? super E> filter)

Iterable interface

- To traverse any collection it provides an Iterator.
- Enable use of for-each loop.
- In java.lang package
- Iterable yeilds an iterator
- Methods
    - Iterator iterator()
    - default Spliterator spliterator()
    - default void forEach(Consumer<? super T> action)

Iterator

- Part of collection framework (1.2)
- Methods
    - boolean hasNext()

- E next()
- void remove()