Generic Methods

```
Static void swap(Object x, Object y){
    Object t = x;
    x = y;
    y = t;
    Sysout("x = "+x+", y = "+y);
}

Static void main(String[] args){
    String s1 = "A", s2 = "B";
    Swap(s1, s2);
      // x = B, y = A
    Integer i = 12;
    Double d = 3.14;
    Swap(i, d);
      // x = 3.14, y = 12
}
```

```
Static <T> void swap(T x, T y){
    T t = x;
    x = y;
    y = t;
    Sysout("x = "+x+", y = "+y);
}
Static void main(String[] args){
    String s1 = "A", s2 = "B";
    Swap(s1, s2); // T = String
      // x = B, y = A
    Integer i = 12;
    Double d = 3.14;
    Swap(i, d); //T = Number
      // x = 3.14, y = 12
    ClsName.<Double>Swap(i, d); //T = Double
}
```

type inference

type given manually

↳ example → Compiler error: Integer ⊁ Double

```
//pre-defined class
class Object {
    //...
    public boolean equals(Object o);
}

class Fraction extends Object {
    private int n,d;
    //...
    public boolean equals(Object o) {
        //...
        Fraction other=(Fraction) o;
        if(n==other.n && d==other.d)
            return true;
        return false;
    }
}
```

```
main()
    Fraction f1=new Fraction(...);
    f1.equals("22/7");
```

Comparable

Comparable is standard for comparing "this" object with the given "other" object.

```
// pre-defined 'interface' (1.4)
interface Comparable {
    int compareTo(Object o);
}
class Fraction implements Comparable {
    private int n, d;
    public double value() {
        return (double) n/d;
    }
    @Override
    public int compareTo(Object o) {
        Fraction other = (Fraction) o;
        if(this.value() == other.value())
            return 0;
        if (this.value() > other.value())
            return +1;
        else
            return -1;
    }
}
```

compareTo() method should return diff between this and other object.

- 0 → if this == other
- +ve → if this > other
- -ve → if this < other

```
main();
    Fraction f1 = new Fraction(10, 2);
    Fraction f2 = new Fraction(20, 2);
    diff = f1.compareTo(f2);
        ↳ -1 (f1 < f2)
    diff = f1.compareTo("9/3");
        ↳ ClassCastException
```

Comparable<T>

Comparable is standard for comparing "this" object with the given "other" object.

```
// pre-defined 'interface' (5.0)
interface Comparable<T> {
    int compareTo(T o);
}
class Fraction implements Comparable<Fraction> {
    private int n, d;
    public double value() {
        return (double) n/d;
    }
    @Override
    public int compareTo(Fraction o) {
        if(this.value() == o.value())
            return 0;
        if (this.value() > o.value())
            return +1;
        else
            return -1;
    }
}
```

compareTo() method should return diff between this and other object.

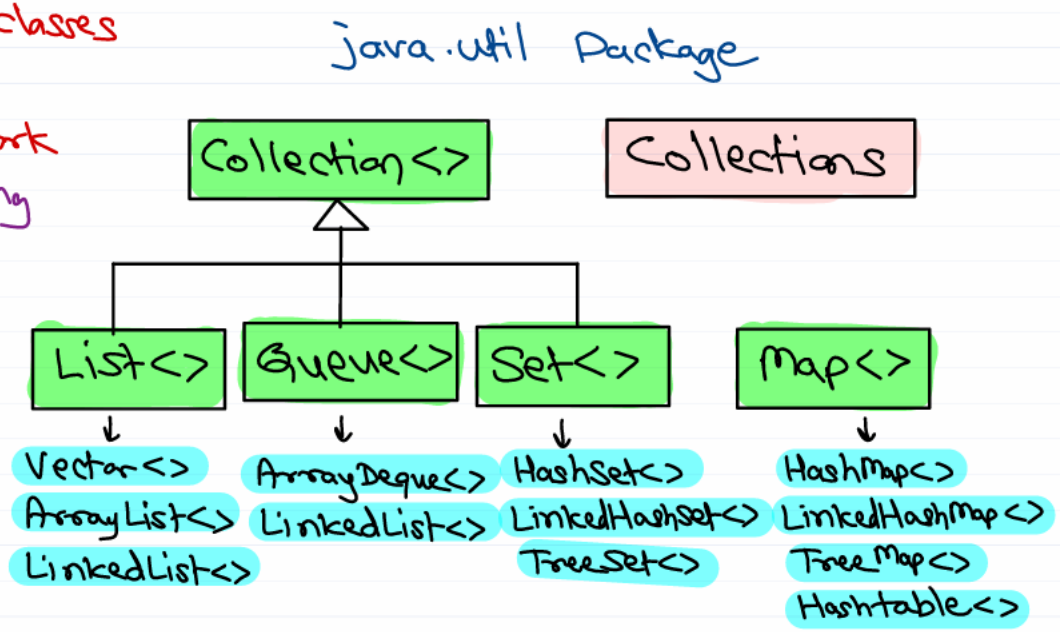
- 0 → if this == other
- +ve → if this > other
- -ve → if this < other

```
main();
    Fraction f1=new Fraction(10, 2);
    Fraction f2=new Fraction(20, 2);
    diff = f1.compareTo(f2); ✓
      ↳ -1 (f1 < f2)
    diff = f1.compareTo("9/3");
                        X
    Compiler error ↵
```

Java Collection Framework

Java 1.0 → Limited collection classes
- ✓ Vector, Hashtable, Stack, ...

Java 1.2 → Collection Framework
- ✓ efficient data processing
  - space & time
- ✓ Collection framework
  - interfaces
  - implementations
  - algorithms



Java collection framework