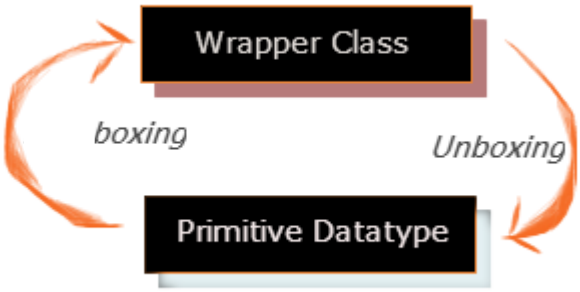
BOXING AND UNBOXING

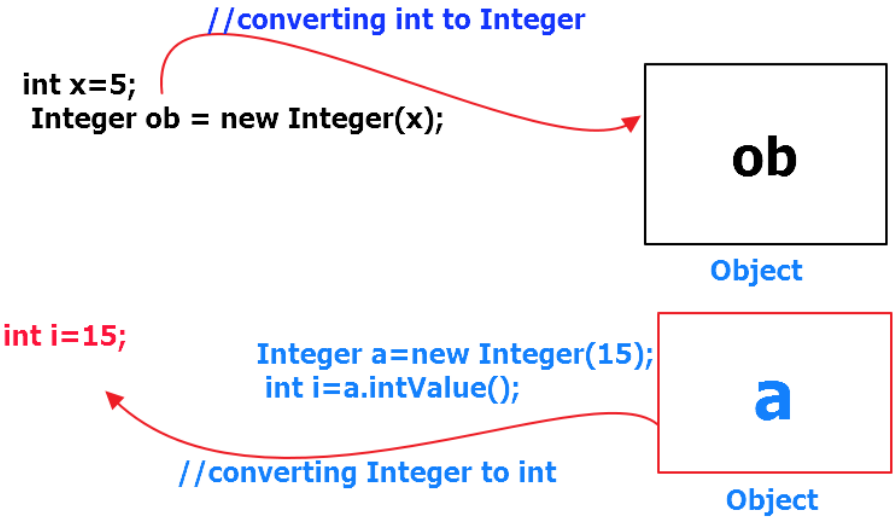
Definition:

Boxing is the **process of converting a primitive data into its object form** using wrapper class.

Unboxing is the process of **converting an object into its primitive form** using primitive data types.

Both boxing and Unboxing can also be automatically performed by compiler.





boxing and Unboxing

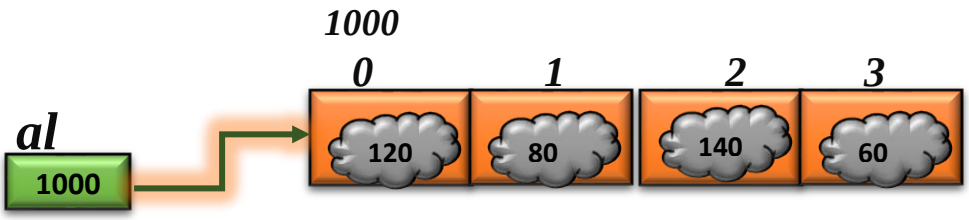
Why should one learn boxing and Unboxing?

*All the collection-**based classes can only store objects** **and not primitive data types. Even if primitive** data **is provided, it is converted into objects** **using** autoboxing **and stored.***

Let us understand this in detail with the help of Arraylist.


```
ArrayList al = new ArrayList();
al.add(120);
al.add(new Integer(120))
al.add(80);
al.add(new Integer(80))
al.add(140);
al.add(new Integer(140))
al.add(60);
al.add(new Integer(60))
```

Internally stored as objects



CODE:

```
import java.util.ArrayList;
class Sis
{
    public static void main(String[] args)
    {
        ArrayList al = new ArrayList();
        al.add(new Integer(120));
        al.add(60);
        al.add(new Double(40.5));
        al.add(80);
        al.add(new Boolean(false));
        al.add("java");
        System.out.println(al);
        Integer a = (Integer)al.get(1);
        System.out.println(a);
    }
}
```

OUTPUT

```
[120, 60, 40.5, 80, false, java]
60
Press any key to continue . . .
```

COLLECTIONS HIERARCHY

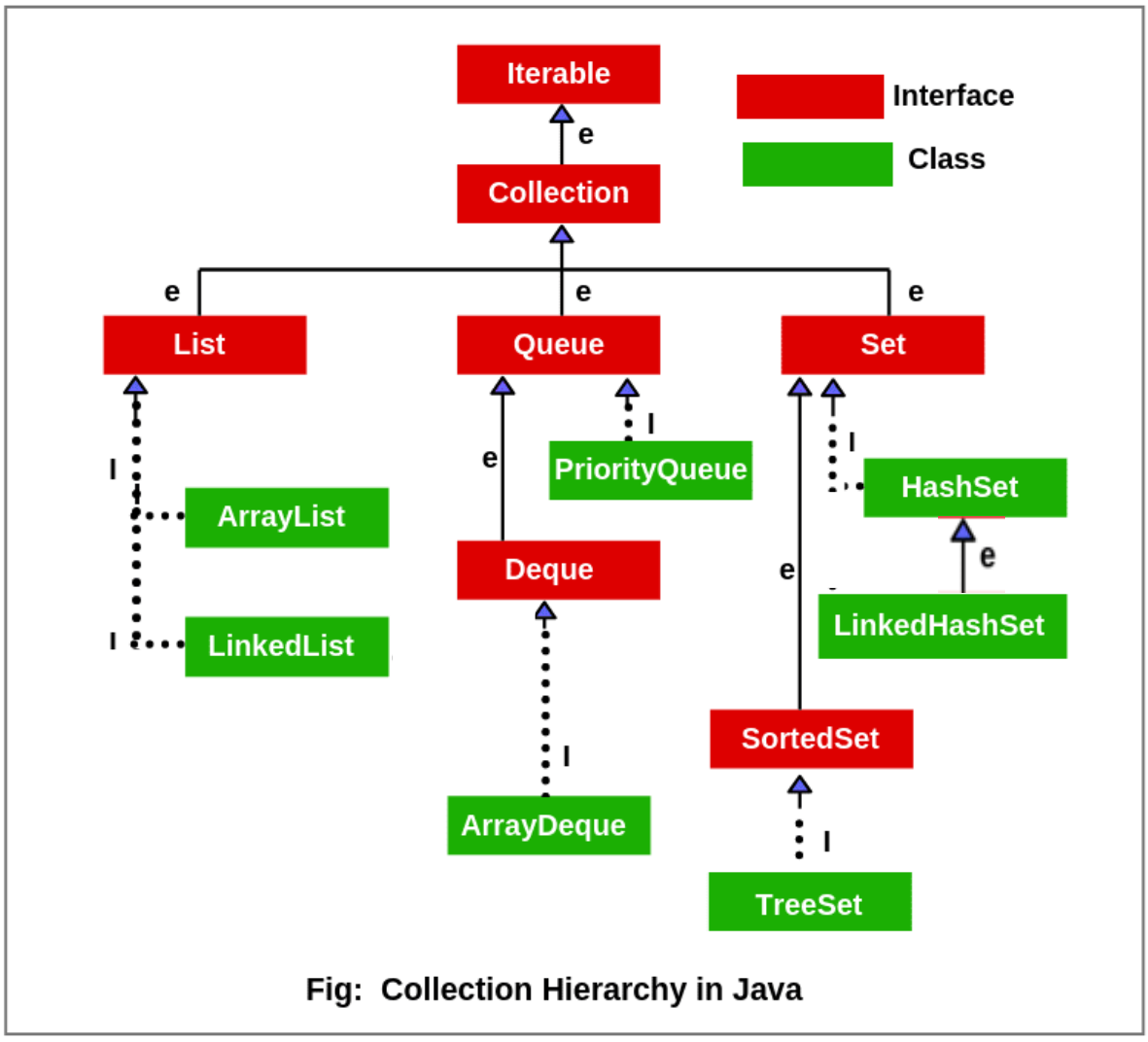


Fig: Collection Hierarchy in Java