**Wrapper Classes:**

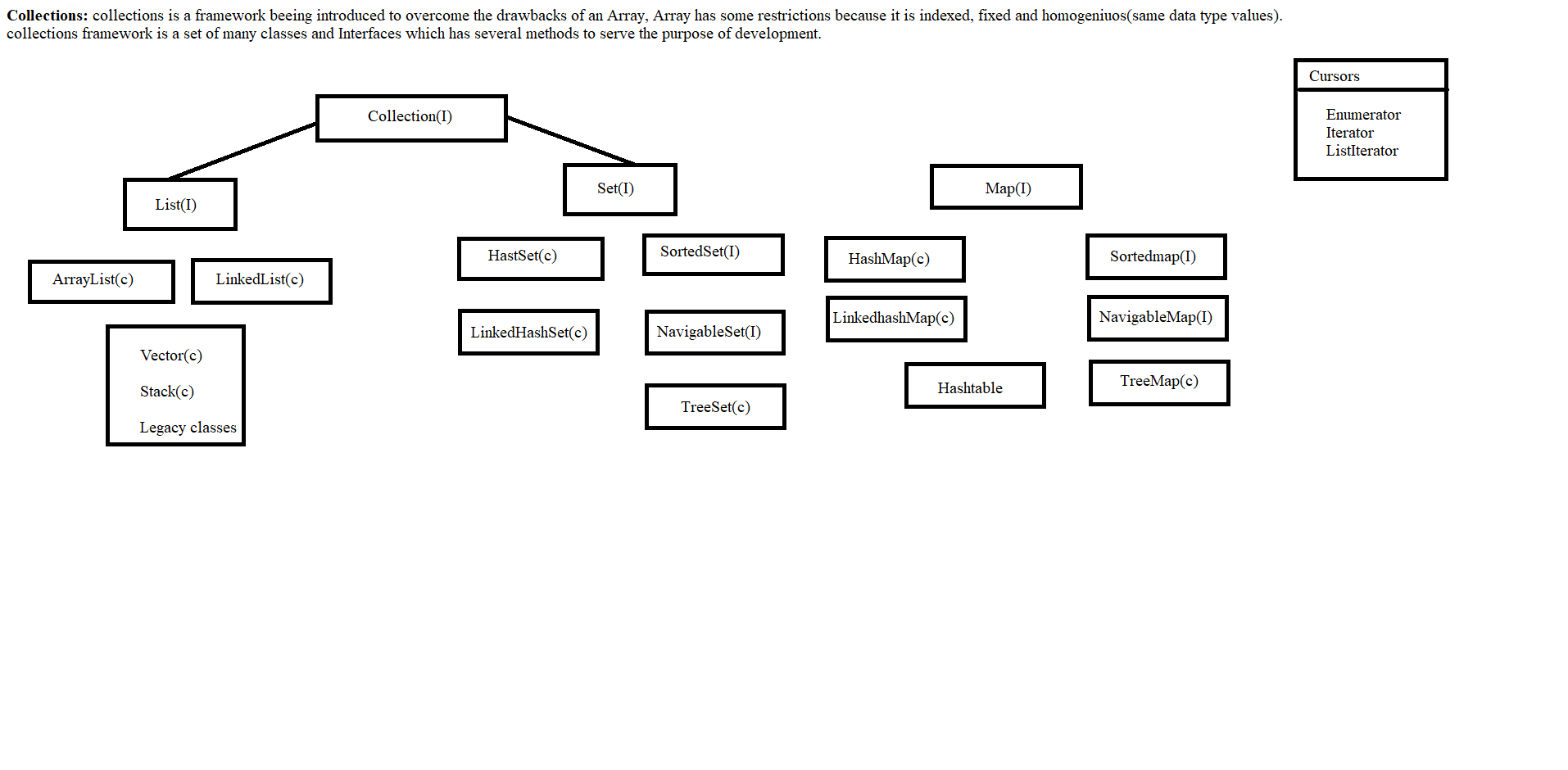
A Wrapper class is a class whose object wraps or contains a primitive data types. When we create an object to a wrapper class, it contains a field and in this field, we can store a primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

**Need of Wrapper Classes:**

* They convert primitive data types into objects. Objects are needed if we wish to modify the arguments passed into a method (because primitive types are passed by value).
* The classes in java.util package handles only objects and hence wrapper classes help in this case also.
* Data structures in the Collection framework, such as ArrayList and Vector, store only objects (reference types) and not primitive types.
* An object is needed to support synchronization in multithreading.
* We use wrapper classes when we are working on frameworks and we use primitive types when we are working on loops . As wrapper classes are slow .

**COLLECTIONS FRAMEWORK:**

Arrays are fixed . in any situation if we want to have a variable collection of elements. As collection is an interface



A Collection is a group of individual objects represented as a single unit. Java provides Collection Framework which defines several classes and interfaces to represent a group of objects as a single unit.

The Collection interface (java.util.Collection) and Map interface (java.util.Map) are the two main “root” interfaces of Java collection classes.

**Need for Collection Framework :**

Before Collection Framework (or before JDK 1.2) was introduced, the standard methods for grouping Java objects (or collections) were Arrays or Vectors or Hashtables. All of these collections had no common interface.

Accessing elements of these Data Structures was a hassle as each had a different method (and syntax) for accessing its members

**LIST INTERFACE**

**package** corejava;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Collections;

**public** **class** Array\_list {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

List array\_list=**new** ArrayList();

/\*

List is an interface, and the instances of List can be created by implementing array list class \*/

array\_list.add(2019);

array\_list.add(2020);

array\_list.add("java");

array\_list.add("collections");

array\_list.add("array list");

//Elements are added irrespective of the data type as we have not specified the type.

System.***out***.println("Elements in the Array List are :" + "\n" +array\_list);