11.1 Concepts Quiz

- Understand the difference between ADT (Abstract Data Types) and specific implementations of such.
- Be able to interpret the Java Collections Framework inheritance hierarchy depicted in Figure 11.1.
- Be able to compare the `ArrayList` and `LinkedList` implementations of the ADT `List`.
- Be able to write code that supports your choice of ADT, but let's you change your mind later on about the specific implementation to use.

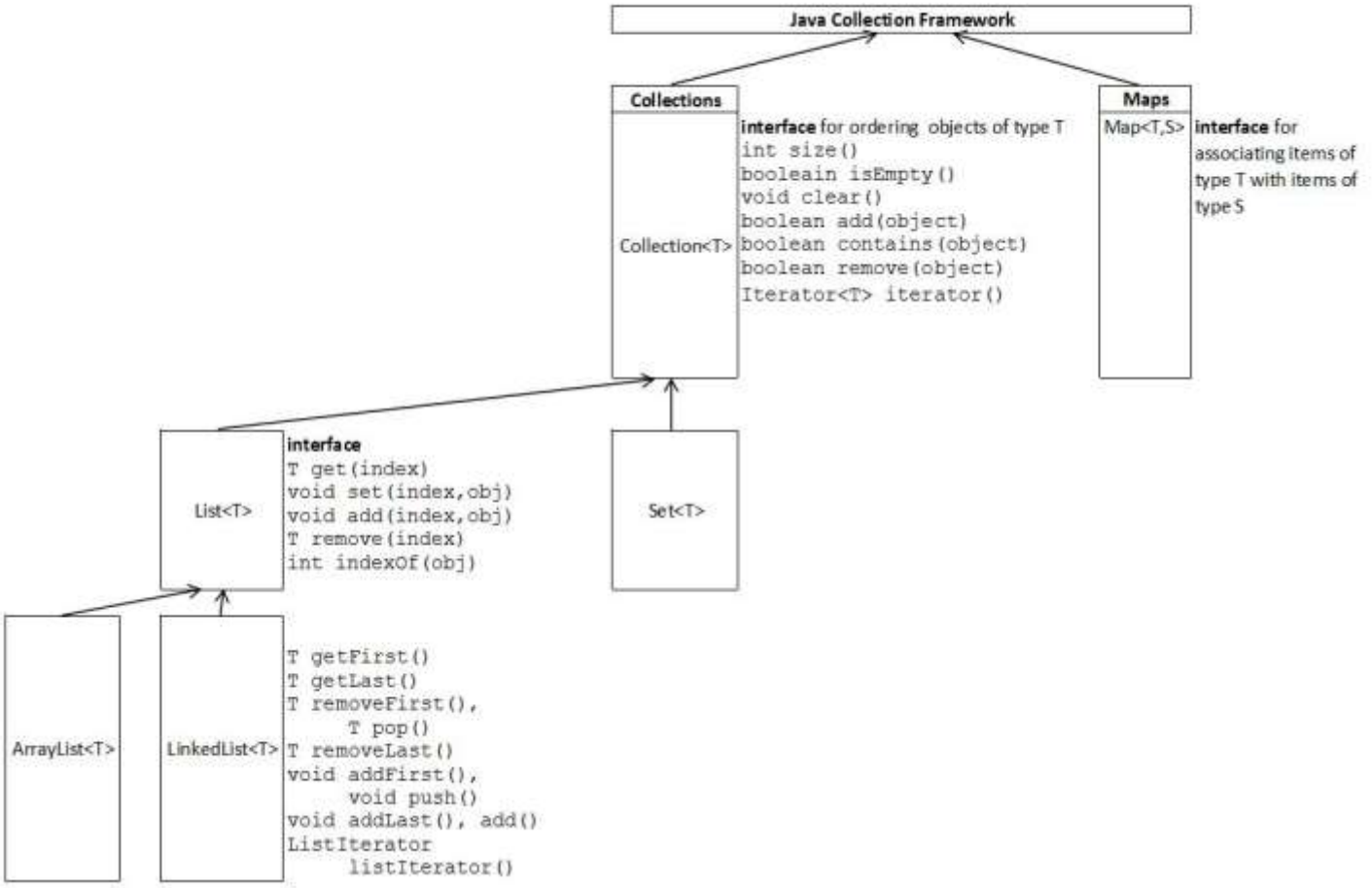
Here are the questions you will be asked:

Which of the following is an ADT (Abstract Data Type?) (Select all that apply.)

- [ ] List
- [ ] Set
- [ ] ArrayList
- [ ] LinkedList
- [ ] Stack
- [ ] Queue
- [ ] Map

`ArrayList` objects and `LinkedList` objects are examples of Java _____ objects. (Select all that apply.)

- [ ] Collection
- [ ] List
- [ ] Set
- [ ] Map



Study the more detailed view of the Java Collections Framework depicted above. When you compare Table 10.1 and Table 10.2 in the textbook to the more detailed JCF, you see that the `ArrayList` class implements fewer additional methods not accounted for by the `List` and `Collection` interfaces than the `LinkedList` class. Why do you suppose `LinkedList` objects have additional methods such as `removeFirst()` and `addFirst()` that `ArrayList` objects lack?

- The `ArrayList` class is designed to discourage these expensive operations.
- `ArrayList` client programs can invoke `add(0,value)` and `remove(0)` methods to obtain the same effect. Why encourage laziness by letting them just say `removeFirst()` or `addFirst(value)`?
- `ArrayList` objects are just naturally less versatile than `LinkedList` objects. It simply isn't possible to perform `removeFirst()` or `addFirst()` operations on them.
- If you google `ArrayList` objects and `LinkedList` objects you see that they have identical methods. So it doesn't really matter which one you choose.

You need to pass a list of GPAs as a parameter to method for processing. Which is the best header for such a method?

- `public void processGPAs(ArrayList<Double> gpaList)`
- `public void processGPAs(LinkedList<Double> gpaList)`
- `public void processGPAs(List<Double> gpaList)`
- `public void processGPAs(Object gpaList<Double>)`

You need a list of GPAs. Which is the best way to declare such a list?

- `ArrayList<Double> gpaList = new ArrayList<Double>();`
- `LinkedList<Double> gpaList = new LinkedList<Double>();`
- `List<Double> gpaList = new ArrayList<Double>();`
- `Object gpaList = new ArrayList<Double>();`