Vectors and ArrayLists have similar features and they both implement the List Interface. They both also handle resizable arrays as the data structure.

The biggest difference when it comes to an ArrayList and a Vector is that a Vector is considered to be synchronized and an ArrayList is not. What this essentially means is that a Vector can only access one thread at a time while an ArrayList can access multiple threads simultaneously. This makes the ArrayList more dynamic in a sense that it can perform multiple tasks such as adding or removing elements at the same time.

As far as performance the ArrayList is considered to be faster due to the fact that it is non-synchronized. Both of them allow for data growth and shrinkage as they are dynamic in sizing. Vectors can use both enumeration and iteration for traversing over elements but ArrayLists can only use iteration.   
  
As a whole, ArrayLists are most often preferred as if needed we can declare an ArrayList to be synchronized like the Vector but we cannot do the same with a Vector and make it non-synchronous.

Stacks are a class in Java that use a stack data structure which is commonly referred to as a last in first out data structure. A common way of thinking about this is stacking blocks on top of each other. You must remove the last placed item before moving onto the next. There are multiple Implemented Interfaces of the Stack class. They are: Serializable, Clonable, Interable<E>, Collection<E>, List<E>, RandomAccess.