Home Assignment-3

1. Differentiate between Array and collection in Java?

Ans:

Arrays	Collection
1. Arrays are fixed in size that is once we create an array we can not increase or decrease based on our requirement.	Collection are growable in nature that is based on our requirement. we can increase or decrease of size.
2. With respect to memory Arrays are not recommended to use	With respect to memory collection are recommended to use
3. With respect to performance Arrays are recommended to use.	With respect to performance collection are not recommended to use.
4. Arrays can hold only homogeneous data types elements.	collection can hold both homogeneous and heterogeneous elements.
5. There is no underlying data structure for arrays and hence ready made method support is not available	Every collection class is implemented based on some standard data structure and hence for every requirements. ready made method support is available being a performance.
6. Arrays can hold both object and primitive	collection can hold only object types but not primitive datatypes.

2100080063 Shabbir Basha

2. Differentiate between ArrayList and Vector in Java?

Sol:

ArrayList	Vector
1. ArrayList is not synchronized	Vector is synchronized
2. ArrayList increments 50% of current array size if the number of elements exceeds from its capacity	Vector increments 100% means doubles the array size if the total number of elements exceeds than its capacity.
3. ArrayList is not a legacy class. It is introduced in JDK 1.2.	Vectory is a legacy class.
4. ArrayList is fast because it is non-synchronized	Vector is slow because it is synchronized, i.e., in a multi-threading environment it holds the other threads in runnable or non-runnable state until current thread releases the lock of the project.
5. ArrayList uses the Iterator interface to traverse the elements	A vector can use the Iterator interface or Enumeration interface to traverse the elements.

2100080063 Shabbir Basha

3. Distinguish between Comparable and Comparator Interface?

Sol:

Comparable	Comparator
1) The Comparable provides a single sorting sequence. In other words, we can sort the collection on the basis of a single element such as id, name, and price	The Comparator provides multiple sorting sequences. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc.
2. Comparable affects the original class, i.e., the actual class is modified	Comparator doesn't affect the original class, i.e., the actual class is not modified.
3. Comparable provides CompareTo() method to sort elements	Comparator provides compare() method to sort elements.
4. Comparable is present in java.lang package	A comparator is present in the java.util package.
5. We can sort the list elements of Comparable type by Collections.sort(List) method.	We can sort the list elements of comparator type by Collections.sort (List, Comparator) method.

2100030063 Shabbir Basha

4. Differentiate between HashMap and Hashtable?

Sol:

HashMap	Hashtable
1. HashMap is non Synchronized. It is not thread safe and can't be shared between many threads without proper Synchronization.	Hashtable is Synchronized. It is thread-safe and can be shared with many threads.
2. HashMap allows one null key and multiple null values	Hashtable doesn't allow any null key or value.
3. HashMap is a new class introduced in JDK 1.2	Hashtable is a legacy class.
4. HashMap is fast	Hashtable is slow
5. We can make the HashMap as Synchronized by calling this code Map m = Collections.synchronizedMap (hashMap);	Hashtable is internally synchronized and can't be unsynchronized.
6. HashMap is traversed by Iterator	Hashtable is traversed by Enumerator and Iterator
7. Iterator in HashMap is fail-fast	Enumerator in Hashtable is not fail-fast
8. HashMap inherits AbstractMap class	Enumerator in Hashtable inherits Dictionary class.

21000 80063 Shabbir Bagha

5. Difference between HashSet and TreeSet in Java?

Sol:	HashSet	TreeSet
1.	It doesnot provide a guarantee to sort the data	It provides a guarantee to sort the data. The sorting depends on the supplied Comparator.
2.	In HashSet, only an element can be null	It doesnot allow null elements
3.	It uses hashCode() or equals() method for comparison.	It uses compare() or compareTo() method for comparision.
4.	It is faster than TreeSet	It is slower in Comparison to HashSet.
5.	Internally it uses HashMap to store its elements	Internally it uses TreeMap to store its elements.
6.	HashSet is backed up by a hash-table	TreeSet is backed up by a Red-black Tree
7.	It allows only heterogeneous value	It allows only homogeneous value.

2100080063 Shabbir Bashe