Collection (Interface)				
SET (Interface)				
Points	HashSet (Class)	Linked Hash Set (Class)	Sorted Set (Interface)	Tree Set (Class)
Implements and Extends	HashSet (C) ----(extends)----> Abstract Set (C) ----(Implements)-----> Set (I) ----(extends)-----> Collection (I)	Linked Hash Set (C) --------(extends)------> Hash Set (C) ------(extends)----> Abstract Set (C) -------(Implements)-----> Set (I) --------(extends)--------> Collections (I)	Sorted Set (I) -------(extends)--------> Set (I) --------(extends)-----> Collections (I)	Tree Set (C) --------(Implements)------> Navigable Set (I) -----------------(extends)-------->Sorted Set (I) -------(extends)-------> Set (I) --------(extends)-----> Collections (I)
Duplicate Allowed	No	No	No	No
Insertion Order	Not Preserved	Preserved	Not Preserved	Not Applicable
Hetrogenious Elements	Allowed	Allowed	-	Not (exception saying classcastexception)
Null Insertion	Allowed (only once)	Allowed (only once)	-	Not (exception saying NullPointerException)
Underline Data Structure	hashtable	hashtable + LinkedList (that is hybrid data structure).	-	balanced tree
Synchronised Methods	No	No	No	No
Thread	Not Safe	Not Safe	Not Safe	Not Safe
Initial Capacity	Initial capacity (16) and load factor (0.75)	Initial capacity (16) and load factor (0.75)	-	0
Version	1.2	1.4	1.2	1.2
Best for which operation	search operation	if we want to develop cache based application		If we want to represent objects according to some sorting order