**Functional Interface:** Exactly one abstract method. provides way to invoke Lambda Expression

**List**

Insertion order is preserved

duplicated allowed

ArrayList, LinkedList (Vector -> stack)

**Set**

duplicates not allowed (adding duplicates, no errors, but only one object is maintained)

order is not preserved

HashSet, TreeSet (useful for sorting)

**Map**

key, value pair

Order not preserved for Hash

**Comparator**

int compare(Object obj1, Object obj2)

obj1 < obj2 --> -ve obj1 come before obj2 in sorting

obj1 = obj2 = 0

obj1 > obj2 --> +ve obj1 come after obj2 in sorting

**All Anonymous classes can't be converted to lambda expressions:**

Can we write Lambda?

Anonymous inner class that extends concrete class. (NO)

Anonymous inner class that extends abstract class. (NO)

Anonymous inner class that implements an interface that contains more than one interface. (NO)

Anonymous inner class that implements an interface that contains Only one interface. (YES)

**this** in anonymous and lambda

Inside anonymous inner class we can declare instance variables. inside lambda we can not only

local variables. and in anonymous inner class this always refers to instance of inner class.

in lambda expression this refers to outer side class members.

**Difference between Anonymous Inner class and Lambda Expressions**

|  |  |
| --- | --- |
| **Anonymous Inner class** | **Lambda Expression** |
| It is a class without name | It is a function without name |
| I can extend Abstract class and concrete class | Lambda can’t |
| Can implement interface with multiple methods | can implement interface with only one method |
| Inside AIC we can declare instance variables | We can’t declare instance variables, what ever variables we declare are local variables |
| AIC can be instantiated | Can not be instantiated |
| in AIC, this always refers to current Anonymous Inner class object but not outer class object | this always refer to outer class object, ie enclosing class object |
| A separate class file will be created when compiled | No separate class. |
| Memory will be allocated on demand whenever, we are creating object | Lambda expression will reside in permanent memory of Jum(Method area) |