**Abstract classes in collection framework**

**1. AbstractCollection:**

AbstractCollection implements most of the Collection interface.

**2. AbstractList:**

AbstractList extends AbstractCollection and implements most of the List interface.

**3. AbstractSequentialList:**

AbstractSequentialList extends AbstractList for use by a collection that uses sequential rather than random access of its elements.

**4. AbstractSet:**

AbstractSet extends AbstractCollection and implements most of the Set interface.

**5. AbstractMap:**

AbstractMap implements most of the Map interface.

Concrete Collection:

| **Collection Type** | **Description** |
| --- | --- |
| ArrayList | An indexed sequence that grows and shrinks dynamically |
| LinkedList | An ordered sequence that allows efficient insertions and removal at any location |
| HashSet | An unordered collection that rejects duplicates |
| TreeSet | A sorted set |
| EnumSet | A set of enumerated type values |
| LinkedHashSet | A set that remembers the order in which elements were inserted |
| PriorityQueue | A collection that allows efficient removal of the smallest element |
| HashMap | A data structure that stores key/value associations |
| TreeMap | A map in which the keys are sorted |
| EnumMap | A map in which the keys belong to an enumerated type |
| LinkedHashMap | A map that remembers the order in which entries were added |
| WeakHashMap | A map with values that can be reclaimed by the garbage collector if they are not used elsewhere |
| IdentityHashMap | A map with keys that are compared by ==, not equals |