Map

* Map is not child interface of Collection Framework.
* Whenever we want to store group of key-value pair we use map.
* Key and value both must be objects
* Values can repeat but not keys.
* All Map implementation class provide two “Standard” constructor:
* A void (no argument) constructor which creates an empty Map.
* A constructor with a single argument of type Map, which creates a new Map with the same Key-value mappings as it’s arguments.

Map.Entry (inner interface):-

* A Map empty (Key-value pair)
* Inner interface of Map.
* Methods of Entry:
* Public object getkey()

It used to obtain key.

* Public object getvalue()

It is used to obtain value.

Useful methods of Map:-

* Public object put(Object key,object value)

It used to insert an entry in this Map.

* Public void putAll(Map map)

It is used to insert the specified map in this Map.

* Public object remove(object key)

It is used to delete an entry for the specified key

* Public object get(Object key)

Return value for specified key.

* Publicboolean containsKey(object key)

Search the specified key in this Map.

* Public Set keySet()

It returns the Set view containing all the Keys

* Public Set emtrySet()

It returns the the Set view containing all keys and values.