**Map-**

Java's Map interface (java.util.Map) can be generified. In other words, you can set the specific type of both the keys and values in a generic Map instance. Here is an example:

Map<Integer, String> set = new HashMap<Integer, String>;

This Map can now only accept Integer instances as keys, and String instances as values.

***import*** *java.util.HashMap;*

***import*** *java.util.Iterator;*

***import*** *java.util.Map;*

***import*** *java.util.Set;*

***public******class*** *MapDemo {*

***public******static******void*** *main(String args[])*

*{*

*Map<Integer, String> map=****new*** *HashMap<Integer, String>();*

*map.put(1, "CDAC-Jaipur");*

*map.put(2, "CDAC-Pune");*

*map.put(3, "CDAC-Mumbai");*

*//now use to map entry for set*

*Set<Map.Entry<Integer, String>> set=map.entrySet();*

*Iterator<Map.Entry<Integer, String>> itr=set.iterator();*

***while****(itr.hasNext()){*

*Map.Entry e=itr.next();//no need to typecast*

*System.****out****.println(e.getKey()+" "+e.getValue());*

*}*

*}*

*}*

**OutPut-**

1 CDAC-Jaipur

2 CDAC-Pune

3 CDAC-Mumbai