**Concurrent Collection– Part\_06**

* **ConcurrentHashMap Example:**

import java.util.concurrent.\*;

import java.util.\*;

class ConcurrentHashMapDemo extends Thread{

static ConcurrentHashMap m = new ConcurrentHashMap();

public void run(){

try{

Thread.sleep(2000);

} catch(InterruptedException ie){

}

System.out.println(“Child thread updating Map”);

m.put(103, “C”);

}

public static void main(String[] args) throws Exception{

m.put(101, “A”);

m.put(102, “B”);

ConcurrentHashMapDemo thread = new ConcurrentHashMapDemo();

thread.start();

Set s1 = m.keySet();

Iterator itr = s1.iterator();

while(itr.hasNext()){

Integer I1 = (Integer) itr.next();

System.out.println(“Main Thread Iterating Map and Current Entry is: “+I1+”…”+m.get(I1));

Thread.sleep();

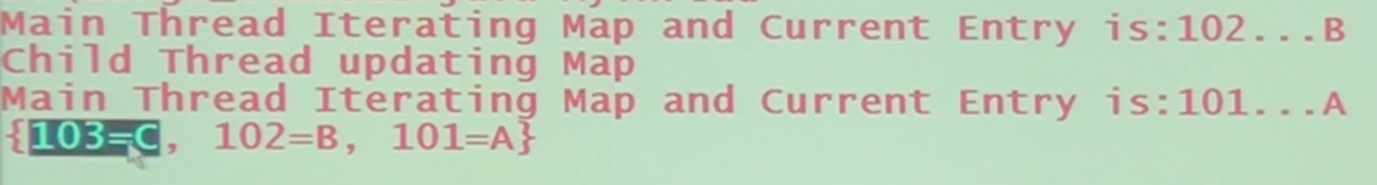
}

System.out.println(m);

}

}

Output:



* **Conclusion:**

While main thread iterating the map, child thread is allowed to update the map, but there is no guarantee that the updated value is available to the iterator, it may be available or it may not be available.