```java
---------ConcurrentModificationTest.java-------
package com.ameya.test;

import java.util.HashMap;
import java.util.concurrent.ConcurrentHashMap;

public class ConcurrentModificationTest  extends Thread{
    //static HashMap<Integer, String> h=new HashMap<Integer, String>();
    static ConcurrentHashMap<Integer, String> h=new ConcurrentHashMap<Integer, String>();
    public void run() {
        try {
            Thread.sleep(1000);
            //Child Thread trying to add Object
            //Adding new Element in the object with key 103
            h.put(103,"Ameya Joshi");
        } catch (InterruptedException e) {
            System.out.println("Child Thread adds Objects");
        }

    }

    public static void main(String[] args) throws InterruptedException{
        h.put(100, "Avani Joshi");
        h.put(101, "Kshiti Joshi");
        h.put(102, "Amol Joshi");
        ConcurrentModificationTest t=new ConcurrentModificationTest();
        t.start();
        for(Object s : h.entrySet()) {
            Object o=s;
            System.out.println(s);
            Thread.sleep(1000);
        }
        System.out.println(h);
    }

}
/*
ConcurrentHashMap Advantages:

```

```
Provides very high concurrency in a multithreaded env.
Read operation can be very fast when the write oprn is done with a lock
It may use multitude of locks
Allows other threads to iterate the objects when one thrad is iterating
Is thread-safe, especially in case of multithreading

Parameters
HashMap                                             ConcurrentHashMap

Synchronization
Non-Synchronized                          Synchronized

Thread Safety                                                Non-Thread
Safe                            Thread safe

Iterator                                                     It is fail-Fast and
throws              It is fail safe and performs iterations
                                                                    exception
during iteration          in multi threaded way

Performance
faster                                              Slower than
HashMap
*/
```