ECAP615

Programming in Java

Harjinder Kaur

Assistant Professor

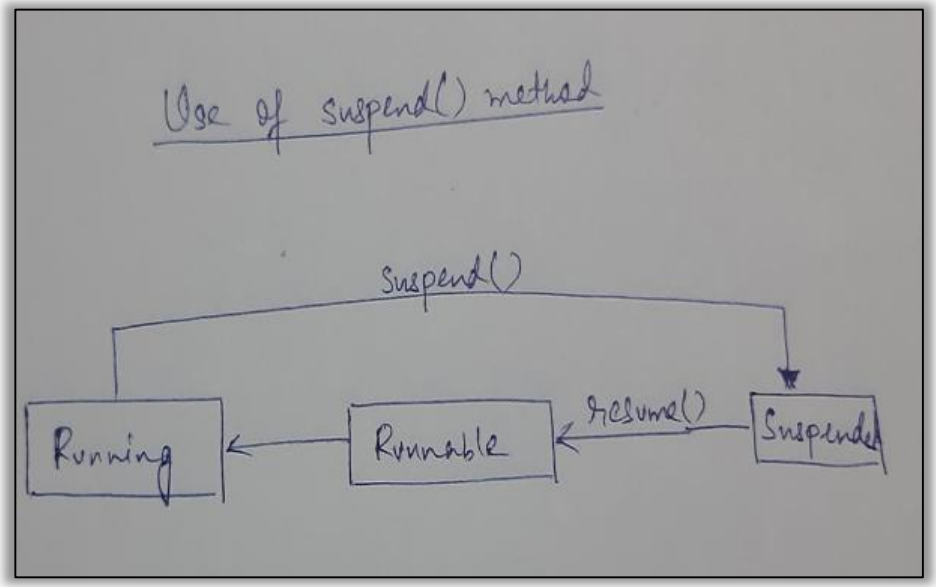
Learning Outcomes

After this lecture, you will be able to

- learn the basic concept suspending and resuming a thread.
- implement suspend() and resume().

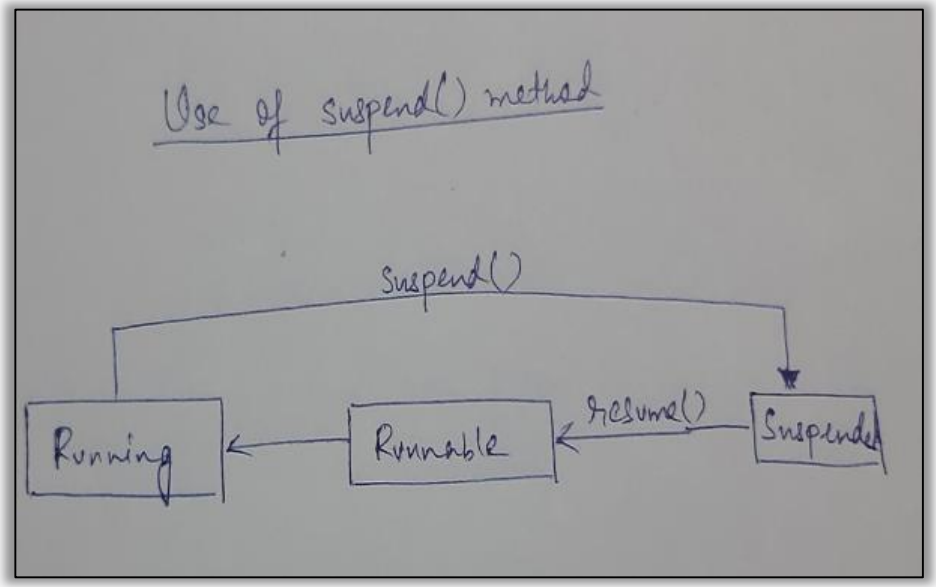
Suspending a Thread

Due to the multi-threading concept, we may need to block one particular thread while some other thread is executing.



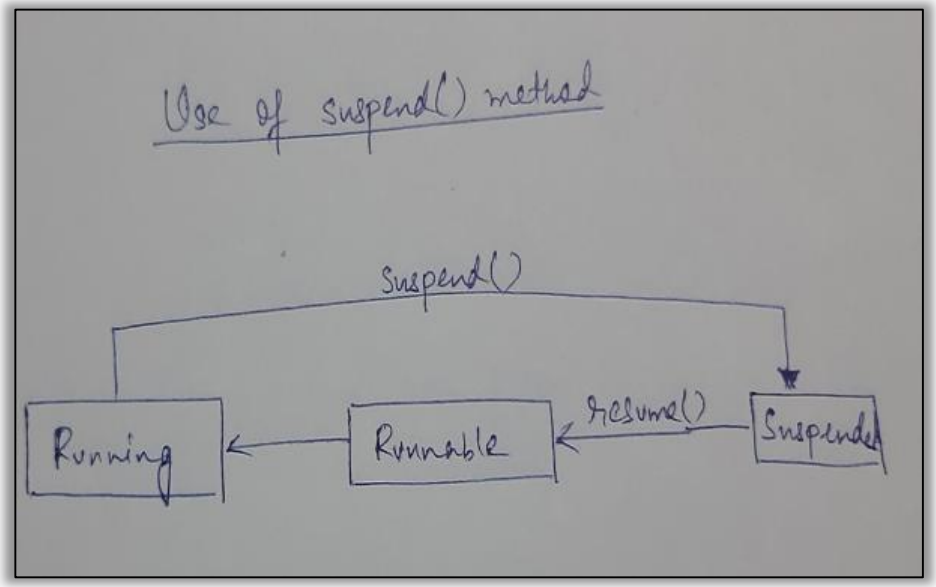
Suspending a Thread

As we have seen in the life cycle diagram, there are several inbuilt methods provided to us by the thread class to change the state of a thread.



Suspending a Thread

- One of them is the suspend() method.
- It is used to change the thread from a Running state to Suspended/Blocked state.



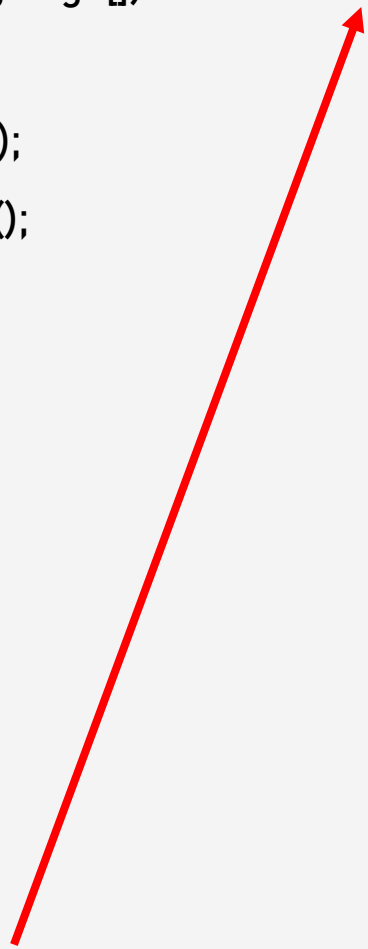
Example

```
public class Testing extends Thread
{
  public void run()
  {
    try
    {
       for(int i=0;i<7;i++)
       {
         Thread.sleep(500);
   System.out.println(this.getName() + ": " + i);
       }     }      catch(InterruptedException e)
    {
       e.printStackTrace();
    }   }
```

Example

```
public static void main(String args[])
 {
   Testing srd1=new Testing();
   Testing srd2=new Testing();
   srd1.setName("First");
   srd2.setName("Second");
   srd1.start();
   srd2.start();
   try
   {
     Thread.sleep(1000);
Thread.sleep(1000);
       srd1.resume();
       System.out.println("Resuming thread
First");

Thread.sleep(1000);
       srd2.suspend();
       System.out.println("Suspending thread
Second");
       Thread.sleep(1000);
       srd2.resume();
       System.out.println("Resuming thread
Second");    }
    catch(InterruptedException e)
    {
      e.printStackTrace();
    }   }  }
```



Resuming a Thread

- Suspend() method puts thread from running to waiting state.
- And thread can go from waiting to runnable state only when resume() method is called on thread.
- Suspend method is deprecated method.

Resuming a Thread

- Resume() method is only used with suspend() method that's why it's also deprecated method.
- Suspend() and remove() are deprecated methods because if not used properly they might lead to deadlock.

Resuming a Thread

```
public class Resume implements Runnable {

    public void run() {

        for (int i = 0; i <=3; i++) {

System.out.println(Thread.currentThread().getName() + " " + i);

        }

    }

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

 {

        Thread th = new Thread(new Resume());

        Thread th1 = new Thread(new Resume());
```

Continue….

Resuming a Thread

Continue….

```
System.out.println("Starting " + th.getName() + "...");
        th.start();
        System.out.println("Suspending " + th.getName() + "...");
  //Suspend the thread.
        th.suspend();
        System.out.println("Starting " + th.getName() + "...");
        th1.start();
        th1.join();
        // Resume the thread.
        th.resume();
System.out.println("Starting " + th.getName() + "...");
    }
}
```


That’s all for now…