Read Me:-

i. Before going through below exercises please visit the link given below, where you can experience the coding standard that each and every developer should follow.

ii. This Code Conventions for the Java Programming Language document contains the standard conventions that Sun follow and recommend that we should follow. It covers filenames, file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices and includes a code example.

iii. LINK - http://www.oracle.com/technetwork/java/codeconv-138413.html

Multithreading Lab Exercise Day-5
Duration: 2 Hours

Program: 1

Inter thread communication without using wait() and notify() methods

```
class Producer extends Thread{
StringBuffer sb;
boolean dataprodover=false;//dataprodover will be true when data production is over
	Producer(){
		sb=new StringBuffer();
	}
	public void run(){
		for(int i=1;i<=10;i++){
			try{
				sb.append("item"+i+"\n");
				Thread.sleep(1000);
				//System.out.println("appending");
			}
			catch(Exception e)
			{
				e.printStackTrace();
			}
		}
		dataprodover=true;
	}
}
class Consumer extends Thread{
	Producer prod;
	Consumer(Producer prod)	{
			this.prod=prod;
		}
		public void run(){
			try{
				while(!prod.dataprodover)
					Thread.sleep(100);
			}
			catch(Exception e){
				e.printStackTrace();
			}
			System.out.print(prod.sb);
		}
}
public class ThreadCommunication1 {
		public static void main(String[] args) {
			Producer obj1=new Producer();
			Consumer obj2=new Consumer	(obj1);
			Thread t1=new Thread(obj1);
			Thread t2=new Thread(obj2);
			t1.start();
			t2.start();
		}
}
```

Program: 2

Inter thread communication by using wait() and notify() methods

```
class Producer1 extends Thread{
		StringBuffer sb;
		Producer1(){
			sb=new StringBuffer();
		}
		public void run(){
			synchronized(sb){
			for(int i=1;i<=10;i++){
				try{
					sb.append("item"+i+"\n");
					}
				catch(Exception e)
				{
					e.printStackTrace();
				}
```

```
                }
                sb.notify();
        }
         }
}
class Consumer1 extends Thread{
        Producer1 prod;
        Consumer1(Producer1 prod){
                this.prod=prod;
        }
        public void run(){
synchronized(prod.sb)//wait till a notification is received from procedure
                {
                try{
                        //while(!prod.dataprodover)
                        prod.sb.wait();
                                //Thread.sleep(100);
                }
                catch(Exception e){
                        e.printStackTrace();
                }
                System.out.print(prod.sb);
        }
        }
}


public class ThreadCommunication2 {
        public static void main(String[] args) {
                Producer1 obj1=new Producer1();
                Consumer1 obj2=new Consumer1(obj1);
                Thread t1=new Thread(obj1);
                Thread t2=new Thread(obj2);
                t2.start();
                t1.start();
        }
}
```

All the Best