**Threads list – 1 – Starting threads**

https://www.youtube.com/watch?v=YdlnEWC-7Wo&list=PLBB24CFB073F1048E

In the tutorial, 3 different ways to start threads are represented in each thread in different packages.

**1st method:**

**package** demo1;

**class** Runner **extends** Thread{

**public** **void** run(){

**for**(**int** i=0;i<10;i++){

System.*out*.print(" N: "+i);

**try** {

Thread.*sleep*(100); //Equals 0.1sec

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

}

**public** **class** App {

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

Runner runner = **new** Runner();

runner.start();

Runner runner1 = **new** Runner();

runner1.start();

/\* Both starts at the same time

But when I wrote "runner.run();, runner1.run();" they took turns in order

\*/

}

}

**2nd method:**

**package** demo2;

**class** Runner **implements** Runnable{

**public** **void** run() {

**for**(**int** i=0;i<10;i++){

System.*out*.print(" N: "+i);

**try** {

Thread.*sleep*(100); //Equals 0.1sec

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

}

**public** **class** App {

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

Thread thread1 = **new** Thread(**new** Runner());

Thread thread2 = **new** Thread(**new** Runner());

thread1.start();

thread2.start();

}

}

**3rd method:**

**package** demo3;

**public** **class** App{

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

Thread t1 = **new** Thread(**new** Runnable(){

**public** **void** run(){

**for**(**int** i=0;i<10;i++){

System.*out*.print(" N= "+i);

**try** {

Thread.*sleep*(100); //Equals 0.1sec

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

});

t1.start();

}

}

**First one extends Thread class.**

**Second one implements Runnable.**

**Third neither extends nor implements.**