**1.a)package** org.virt.com;

**public** **class** RunnabletT **implements** Runnable

{

**private** Thread t;

**private** String threadName;

RunnabletT(String name)

{

threadName = name;

System.***out***.println("creating"+threadName);

}

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

System.***out***.println("Running"+threadName);

**try** {

**for**(**int** i = 4;i>0;i--) {

System.***out***.println("Thread:--"+threadName+" , "+i);

Thread.*sleep*(100);

}

} **catch**(InterruptedException e){

System.***out***.println("Thread"+threadName+"interrupted");

}

System.***out***.println("Thread"+threadName+"Exiting");

}

**public** **void** start() {

System.***out***.println("Starting"+threadName);

**if**(t==**null**) {

t = **new** Thread(**this**,threadName);

t.start();

}

}

}

**b)package** org.virt.com;

**public** **class** Demot

{

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

RunnabletT R1 = **new** RunnabletT("Thread-1");

R1.start();

RunnabletT R2 = **new** RunnabletT("Thread-2");

R2.start();

}

}

2.diff. between extends class and runnable interface

A.extends from thread class

step1.inherit from thread class

step2.override run()

step3.code should be implemented in run()

step4.create object for our class and start thread

step5.start has 2 functionalities

* It sends request to thread scheduler to ask separate thread
* Invoke run() and it starts execution in separate thread.

B.implementing runnable interface

Step1.implement runnable interface and it has only run(),so we have to override run()

Step2.provide logics in run() and execute in separate thread

Step3.create thread class object and pass runnable type object as input for thread class object

Step4.Thread will start thread in runnable interface.