**Multithreading extending thread class**

The procedure for creating threads based on extending the Thread is as follows:

1. A class extending the Thread class overrides the run() method from the Thread class to define the code executed by the thread.

2. This subclass may call a Thread constructor explicitly in its constructors to initialize the thread, using the super() call.

3. The start() method inherited from the Thread class is invoked on the object of the class to make the thread eligible for running.

**Example**

**MultiThreadingDemo.java**

**class** Demo **extends** Thread {

@Override

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

// **TODO** Auto-generated method stub

System.***out***.println(Thread.*currentThread*().getName());

}

Demo() {

// **TODO** Auto-generated constructor stub

}

Demo (String str)

{

**super**(str);

System.***out***.println(**this**);

start();

}

}

**public** **class** MultiThreadingDemo {

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

Thread thread1=**new** Thread(**new** Demo(),"CDAC at Jaipur");

Thread thread2=**new** Thread(**new** Demo(),"CDAC at Pune");

Thread thread3=**new** Thread();

thread1.start();

thread2.start();

thread3.start();

**try** {

Thread.*currentThread*().*sleep*(2000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

};

}

}

**Output**

CDAC at Jaipur

CDAC at Pune