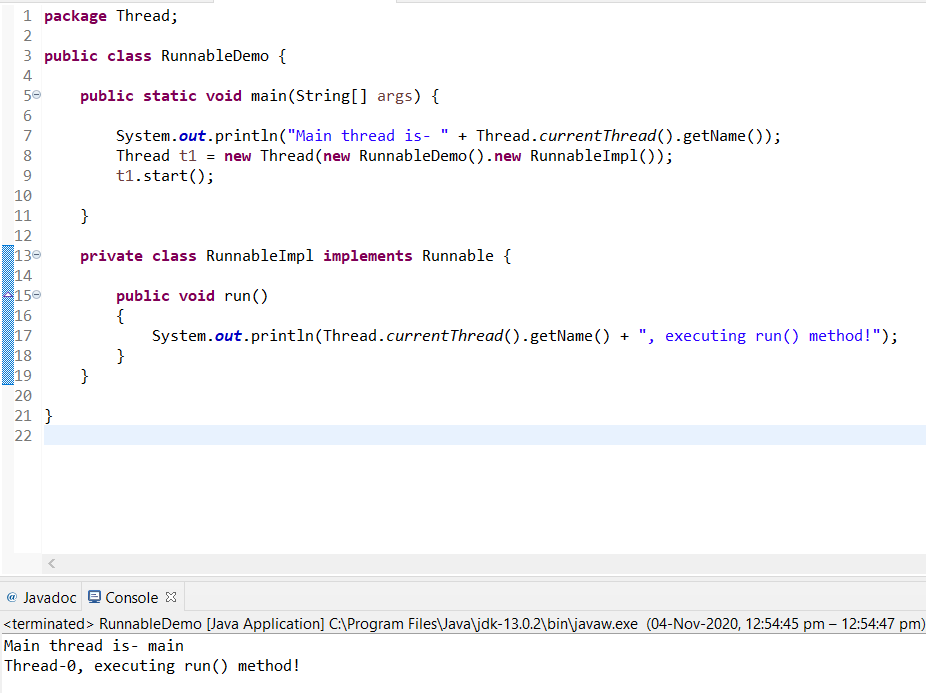


* currentThread() is used to create the reference of the current main thread.
* Then we are printing the current thread.
* We are renaming the current thread to “My Thread” using “setname()” method of the thread class and printing it.
* Now using the for loop, we are decrementing the number 5 and printing it each time.
* “sleep(1000)” method is used to make the CPU idle for specific time. Time is given as a parameter to the sleep method in milli seconds.
* Exception Handling is used to catch errors.



**Thread.currentThread().getName()**

Prints the Name of the Thread.

**RunnableImpl implements Runnable**

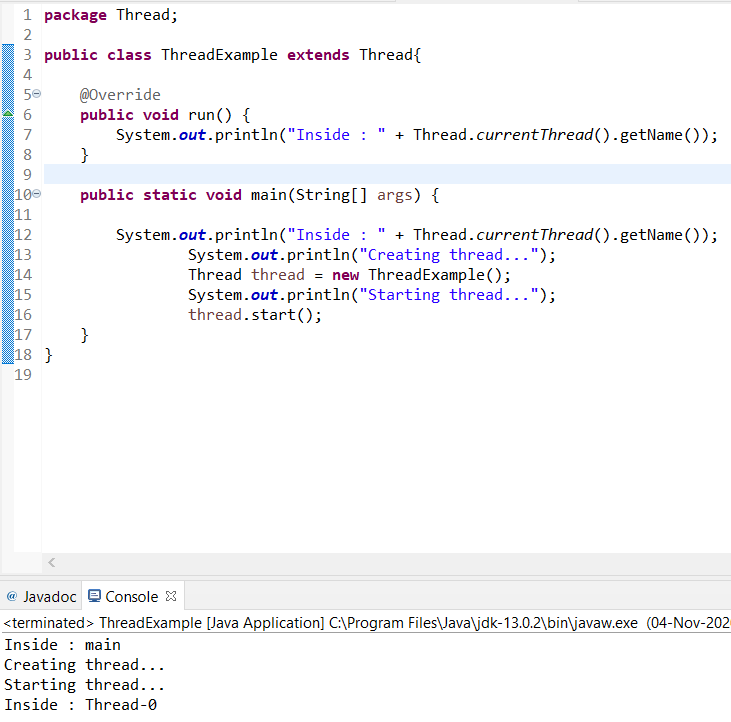
Creates the class “**RunnableImpl**“ implemented from the runnable Interface.

**Thread t1 = new Thread(new RunnableDemo().new RunnableImpl());**

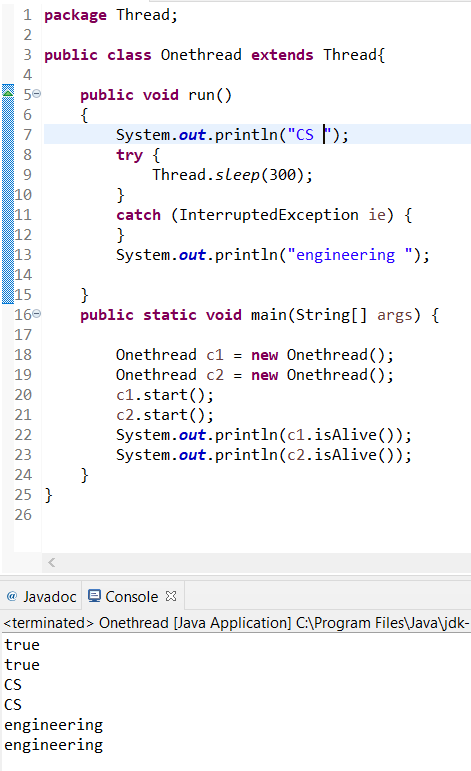
It Instantiates the Thread class and pass the implementer to the Thread, Thread has a constructor which accepts Runnable instance.

**t1.start()**

Starts the implementation of the thread object t1.



* Here Thread Is created by extending from the Thread class.
* Here the run is being over rided and then implemented with our own source code.
* In main we are printing the “Creating Thread” and then instantiating the thread created to a object “thread”.
* Now we are printing “Starting Thread…” and we have called the “start()” method which is predefined in the “Thread” Class.
* The CPU starts the thread implementation and prints the name of the thread.



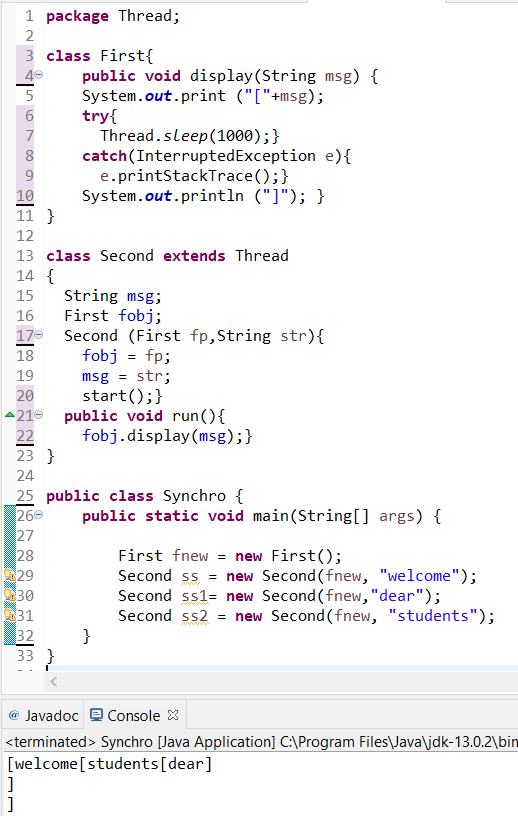
* Here there are three threads.
* One main thread and two child threads.
* “isAlive()” is used to check whether the thread is still alive or not.
* It prints “True” because they are still alive.
* And after the “start()” is called they CPU start the implementation of both the threads.
* The first child thread prints “CS”.
* Since there is a sleep of 300ms after the print statement context switching happens and goes to the second child thread and prints the “CS”.
* Here also there is a sleep(300).
* So again, context switching happens and prints “Engineering”.
* Then happens and prints the Engineering of child thread 2.



“join()” is used to wait until the thread its joining with is completes.

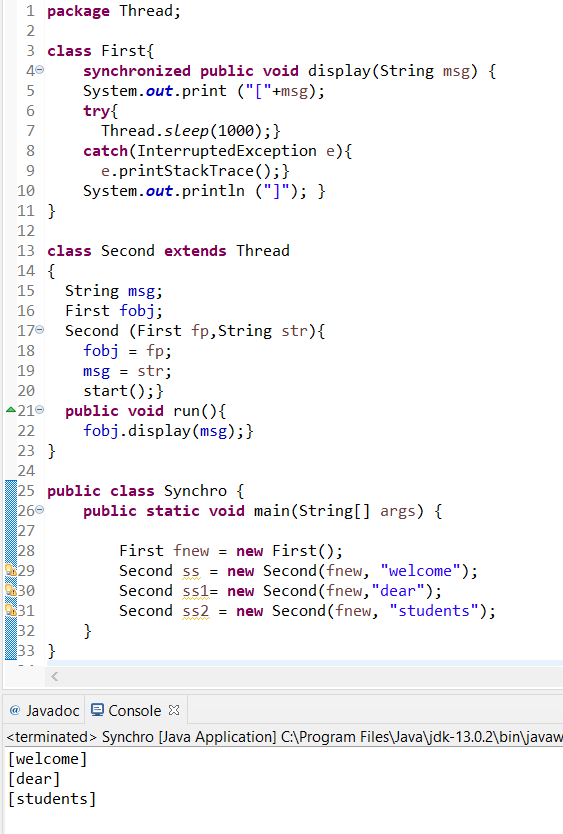
So here in the first child thread both the print statements are executed.

Then the CPU goes with the child thread 2 and prints the 2 statements again.

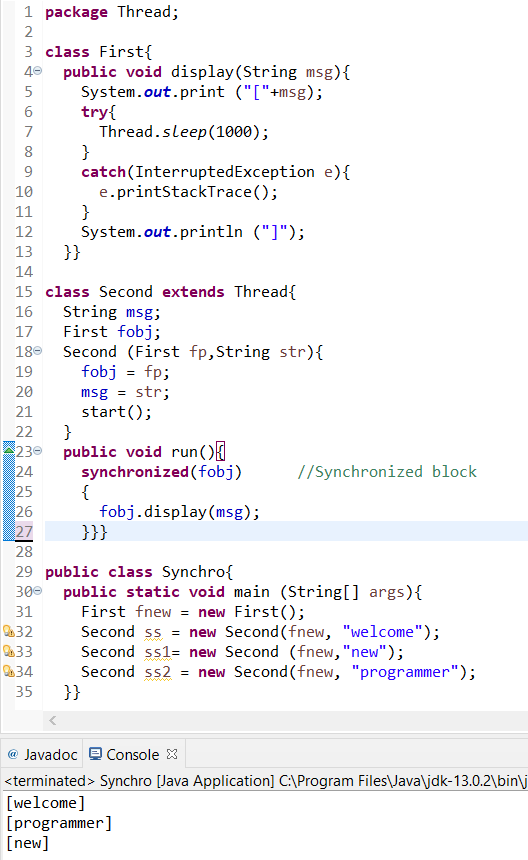


This Program is not Properly Synchronized.

And it doesn’t provide the expected result. To over come this we need to use “Synchronized” keyword along with the method.



Here it prints the messages in proper order in proper square brackets because of “Synchronized” keyword.



Here the “Synchronized” keyword is given in the run() method which calls the thread to start its computation.