Java
Thread APIs (THI)

Mario Alviano

University of Calabria, Italy

A.Y. 2016/2017

Do not invoke Thread.run()

- ✗ The method will be called by the current thread
- ✓ Use `start()`
- ✓ Exception: you may call `run()` for debugging

```
https://www.securecoding.cert.org/confluence/
pages/viewpage.action?pageId=35782697
```

Do not use Thread.stop() to terminate threads

- ✗ Invariants cannot be preserved by abrupt termination
- ✓ Use a boolean volatile flag in a loop
- ✓ Call `interrupt()`

```
https://www.securecoding.cert.org/confluence/
display/java/THI05-J.+Do+not+use+Thread.stop%28%
29+to+terminate+threads
```

Ensure that threads performing blocking operations can be terminated

- ✗ Some I/O operations may block your program
- ✓ Use I/O operations that can be interrupted

```
https://www.securecoding.cert.org/confluence/
display/java/THI04-J.+Ensure+that+threads+
performing+blocking+operations+can+be+terminated
```

Notify all waiting threads rather than a single thread

- ✗ Avoid `notify()` and `signal()`
- ✓ Prefer `notifyAll()` and `signalAll()`

```
https://www.securecoding.cert.org/confluence/
display/java/THI02-J.+Notify+all+waiting+threads+
rather+than+a+single+thread
```

Always invoke wait() and await() methods inside a loop

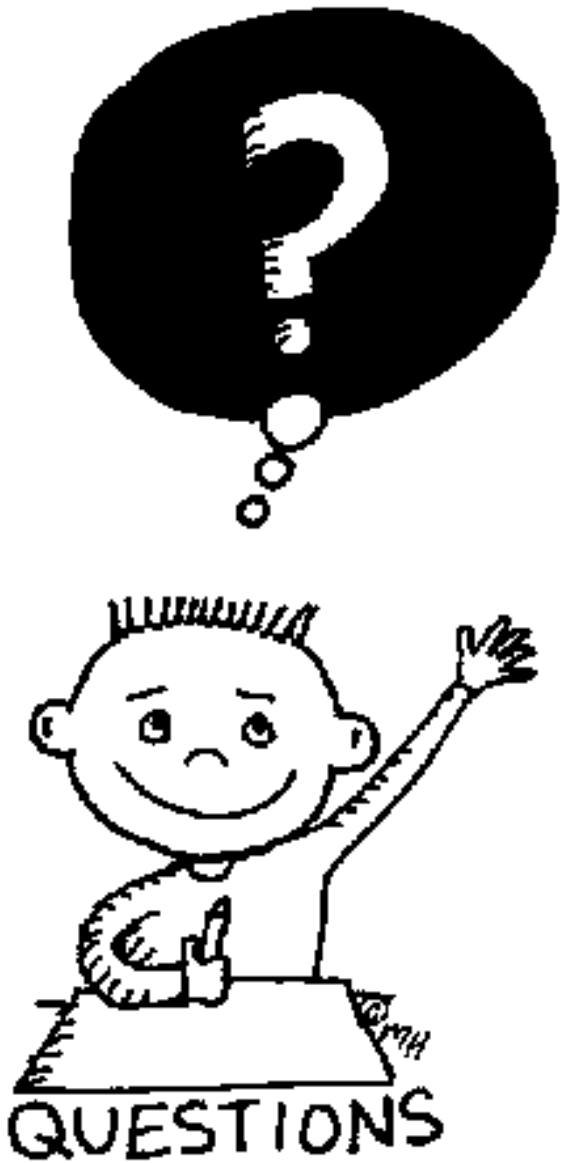
- ✗ You are waiting for something, don't trust external notification
- ✓ The resource may become busy again
- ✓ Avoid time-of-check, time-of-use (TOCTOU) vulnerabilities

```
https://www.securecoding.cert.org/confluence/
display/java/THI03-J.+Always+invoke+wait%28%29+
and+await%28%29+methods+inside+a+loop
```

Do not invoke ThreadGroup methods

✗ They are not thread-safe

```
https:
//www.securecoding.cert.org/confluence/display/
java/THI01-J.+Do+not+invoke+ThreadGroup+methods
```



END OF THE LECTURE